

Veli-Matti Pukkinen

Robottisolun simulointi

Opinnäytetyö

Kevät 2011

Tekniikan yksikkö, Seinäjoki

Tietotekniikka

Mekatroniikka



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Mekatroniikka

Tekijä: Veli-Matti Pukkinen

Työn nimi: Robottisolun simulointi

Ohjaaja: Martti Lehtonen

Vuosi: 2011

Sivumäärä: 41

Liitteiden lukumäärä: 0

Tämän opinnäytetyön tarkoituksena oli simuloida ABB:n robotin solu käytettäväksi RobotStudio-opetusohjelmassa. Solu sijaitsee Seinäjoen ammattikorkeakoulun Tekniikan yksikössä, robotiikan laboratoriossa.

Offline-ohjelmointi eli etäohjelmointi on tuotannon ulkopuolella tapahtuvaa robotin uudelleenohjelmointia, joka on yleistynyt huomattavasti 2000-luvun aikana teollisuudessa. Offline-ohjelmoinnin suurimpina etuina voidaan pitää sitä, että uusia ratkaisuja voidaan testata simuloimalla ennen käyttöönottoa, eikä robotin toimintaa tarvitse keskeyttää uudelleenohjelmoinnin aikana.

Opinnäytetyön pohjana oli Tapio Hellmanin vuosina 2000 - 2002 CAD-ohjelmalla piirretty kuva robotiikan laboratoriosta, josta erotettiin ABB:n robotin ympäristö ja työkalut. Nämä kappaleet käännettiin ACIS-tiedostomuotoon, jotta ne saatiin vietyä RoboStudio-ohjelmaan.

ABB-robotin solu sisältää huoneen seinät, robotin jalustan, työkaluaseman ja robotin työkalut. Nämä siirrettiin Import geometry -toiminnon avulla RobotStudio-ohjelmaan. ABB:n solun sisältämät kappaleet jouduttiin asemoimaan uudelleen robotin maailmankoordinaatiston suhteen. Työkalut asemoitiin vielä uudelleen robotin laipan koordinaatiston suhteen. Työkalut pystytään nyt liittämään robottiin.

Avainsanat: Offline- ohjelmointi, robotiikka, robottisolu, RobotStudio

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Mechatronics

Author: Veli-Matti Pukkinen

Title of thesis: Simulation of a robot cell

Supervisor: Martti Lehtonen

Year: 2011

Number of pages: 41

Number of appendices: 0

The aim of this thesis was to simulate the ABB robot cell for the RobotStudio-program. The ABB robot cell is located in Seinäjoki University of Applied Sciences, School of Technology.

In the 21st century offline programming, which takes place outside production, has become common in industry. It has many benefits, such as testing of new solutions and the fact that robots do not have to stop working during reprogramming.

In 2000 – 2002 Tapio Hellman created a CAD drawing of the robotics laboratory where the robot cell is located. This drawing was used as basis for the thesis. The robot cell had to be separated from the CAD drawing because the entire drawing was too large to be transferred. The robot cell had to be also converted to ACIS-file format.

The robot cell includes the walls of the cell, a robot stand, a tool station and tools for the robot. These objects were transferred to the RobotStudio-program and positioned according to the world coordinate system. Now it is possible to attach tools to the robot.

Keywords: offline programming, robotics, robot cell, RobotStudio

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ	4
Kuvio- ja taulukkoluettelo.....	5
1 JOHDANTO.....	8
1.1 Työn tarkoitus ja tavoite	8
1.2 Työn rakenne	8
2 ROBOTIIKASTA.....	9
2.1 Robotiikka	9
2.1.1 Sovelluskohteet.....	10
2.1.2 Robottijärjestelmä	11
2.1.3 Robottityypit	12
2.1.4 Teollisuusrobotti.....	15
2.1.5 Robotin työkalut	18
2.1.6 Robotin koordinaatistot	19
2.2 Robotin ohjelmointi.....	20
2.2.1 Online-ohjelmointi	20
2.2.1 Offline-ohjelmointi	21
2.3 ABB RobotStudio -ohjelma.....	25
3 TYÖN TOTEUTUS	27
3.1 Kuvan muokkaus	27
3.2 Robotstudio-ohjelman käynnistys.....	27
3.3 Robottisolun siirto ohjelmaan	32
3.4 Työkalujen liittäminen robottiin	35
4 YHTEENVETO	39
LÄHTEET	

Kuvio- ja taulukkoluetelo

Taulukko 1. Teollisuusrobottien määrän kasvu vuosina 1992 - 2005	10
Kuva 1. Teollisuusrobotin komponentit.	11
Kuva 2. Yleisimmät robottityypit.	12
Kuva 3. Portaalirobotti.	13
Kuva 4. Sylinterirobotti.	13
Kuva 5. Scara-robotti	14
Kuva 6. Kiertyvänivelinen robotti.....	14
Kuva 7. Robottityyppien jakauma vuonna 2002	15
Kuva 8. AC- ja DC-servomootorit.....	16
Kuva 9. Pulssiantureita	16
Kuva 10. AC-servomoottorin ohjain	17
Kuva 11. Robotin koordinaatistot	20
Kuva 12. Robotin käsiohjain	21
Kuva 13. IGRIP-ohjelmointi ja simulointiohjelmisto	22
Kuva 14. RAPID-ohjelmointiesimerkki.....	23
Kuva 15. Koordinaattien selitykset	23

Kuva 16. Virtual FlexPendant	24
Taulukko 2. Ohjelmointimenetelmät.....	24
Kuva 17. RobotStudio-ohjelman ABB-library ja Robottimalli IRB 120.....	25
Kuva 18. Robottimalli IRB7600 ja rullakuljetin.....	26
Kuva 19. RobotStudion alkuvalikko.....	28
Kuva 20. Robottimallin valitseminen.	29
Kuva 21. Robottimalli IRB2400.	29
Kuva 22. Jog Joint -toiminto.....	30
Kuva 23. Robotin nivelten liikuttaminen Jog Jointilla.....	30
Kuva 24. Jog Linear -toiminto.	31
Kuva 25. Robotin nivelten liikuttaminen Jog Linearilla.....	31
Kuva 26. Import Geometry -toiminto.	32
Kuva 27. Robotin jalustan ja solun liittäminen.....	33
Kuva 28. Koordinaattien muuttaminen.	34
Kuva 29. Työkaluaseman siirto.....	34
Kuva 30. Robotin työkalut.....	35
Kuva 31. Robotin laipan valinta set local origin -toiminolla.....	37

Kuva 32. Robotin työkalun valinta set local origin -toiminnolla. 38

Kuva 33. Työkalu kiinnitettynä robottiin..... 38

1 JOHDANTO

1.1 Työn tarkoitus ja tavoitteet

Seinäjoen ammattikorkeakoulun Tekniikan yksikössä robotiikan opetuksessa käytetään ABB:n teollisuusrobottimallia IRB2400 ja ABB:n kehittämää RobotStudio-ohjelmistoa. RobotStudio-ohjelmistoa käytetään teollisuusrobotin ohjelmointiin ja erilaisten ratkaisujen simulointiin.

Tämän opinnäytetyön tavoitteena on mallintaa Tekniikan yksikön robotiikan laboratoriossa sijaitsevan ABB:n robotin solu, että sitä voidaan hyödyntää RobotStudio-opetusohjelmassa. Toisena tavoitteena on saada solun sisältämät mallinnetut työkalut liitettyä robottiin RobotStudio-ohjelmassa.

1.2 Työn rakenne

Opinnäytetyön ensimmäinen osio käsittää työn tarkoituksen ja tavoitteet sekä rakenteen. Toinen osio sisältää robotiikan perusteet ja robottien ohjelmointitavat. Kolmas osio sisältää työn toteutuksen ja neljäs osio työn yhteenvedon.

2 ROBOTIIKASTA

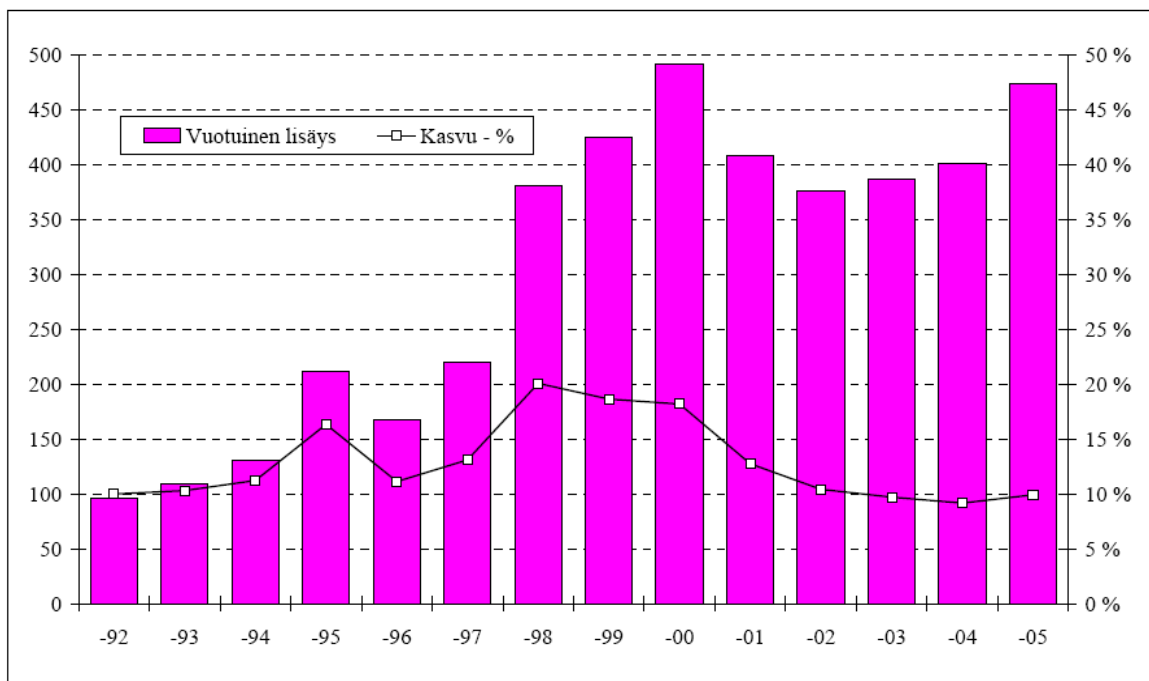
2.1 Robotiikka

Kansainvälinen robottiyhdistys on määritellyt sanaston teollisuusroboteille ja myös luokitellut yleisimmät robottimallit rakenteen mukaan. Standardi ISO 8373 määrittää robotin olevan ohjelmoitava, monipuolisesti toimiva mekaaninen laite, jossa on vähintään kolme niveltä. Käsitteellä robotiikka tarkoitetaan oppia robottien suunnittelusta, rakentamisesta ja soveltamisesta. (Aalto ym, 1999, 12-13.)

Erilaisten työkalujen ja ohjelmoinnin avulla robotin tulee toteuttaa monenlaisia tehtäviä. Robotin liikerata voidaan määritellä etukäteen toimintaympäristön tapahtumien mukaan tai antureiden avulla. Robotteja voidaan käyttää teollisuudessa useissa vaativissa tehtävissä, joissa vaaditaan erittäin tarkkaa liikettä tai isojen kohteiden siirtämistä paikasta toiseen. Myös vaikeat tehtävät, jotka saattavat olla ihmisterveydelle vaaraksi, voidaan usein antaa robottien tehtäväksi. (Keinänen ym. 2000, 307.) 1960-luvulla teollisuusrobotteja pyrittiin hyödyntämään monipuolisesti erilaisissa tehtävissä, mutta vuosikymmenien kuluessa erikoisrobotteja myös ilmaantui teollisuuden käyttöön. (Aalto ym, 1999, 13.)

Nykypäivänä robottien luotettavuus ja työn laatu ovat parantuneet. Tähän ovat vaikuttaneet tekniikan ja ohjelmistojen kehittyminen. (Keinänen ym. 2000, 308.)

Taulukko 1. Teollisuusrobottien määrän kasvu vuosina 1992 - 2005. (Suomen Robotiikkayhdistys Ry 2006.)



2.1.1 Sovelluskohteet

Perinteisiä robottien sovellusalueita tuotantotehtävissä ovat hitsaus, kappaleiden siirto sekä erilaiset kokoonpano- ja pintakäsittelytehtävät (Keinänen ym. 2000, 309).

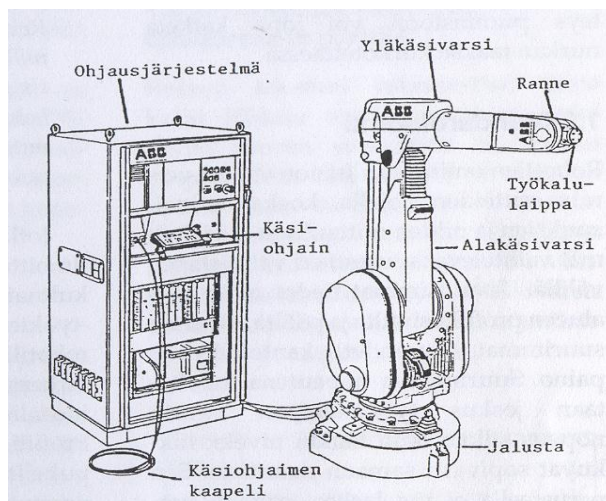
Hitsaus. Yleisimpiä robottien käyttökohteita ovat hitsaustehtävät. (Keinänen ym. 2000, 309.)

Kappaleiden siirto. Kappaletavara-automaatiossa robotit ovat käytössä erilaisissa pakkaustehtävissä ja kappaleiden siirtämisessä tuotannon aikana. (Keinänen ym. 2000, 309.)

Kokoonpano- ja pintakäsittelytehtävät. Elektroniikkateollisuudessa robotteja käytetään erilaisissa kokoonpanotehtävissä. Pintakäsittelytehtäviä ovat esimerkiksi hionta, kiillotus ja maalaus. (Keinänen ym. 2000, 309.)

2.1.2 Robottijärjestelmä

Robottijärjestelmä koostuu useista komponenteista. Luonnollisesti näkyvin osa robotissa on käsivarsi. Siihen liitettävä työkalu suorittaa varsinaisen työn. Kaikkein merkittävimpänä komponenttina voidaan kuitenkin pitää ohjausjärjestelmää, koska se ohjelmistoinen teettää työtä enemmän kuin käsivarsi. Teollisuusrobottisolussa ohjausjärjestelmän hinta on myös korkeampi kuin käsivarren. Muita merkittäviä komponentteja ovat ympäristöä mittaavat anturit, oheislaitteet ja myös kaapelointi. (Aalto ym. 1999, 15.)



Kuva 1. Teollisuusrobotin komponentit (Aalto ym. 1999.)


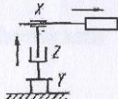


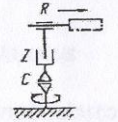


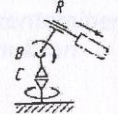

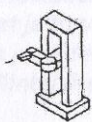
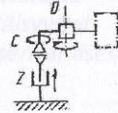


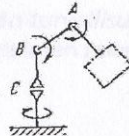


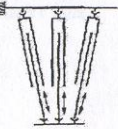

Robotin mekaaniseen rakenteeseen kuuluvat tukivarret ja nivelet. Tukivarret muuttavat asentoaan ja asemiaan nivelten avulla. Robotin perusliikkeitä kuvaa termi vapausaste (DOF, Degree of Freedom). Vapausasteet voivat teollisuusroboteissa olla joko kiertyviä tai suoria. Yleensä vapausaste käsittää yhden toimilaitteen, kuten moottorin tai sylinterin. Poikkeuksena on differentiaaliranne, jossa on on kaksi toimilaitetta vapausastetta kohden. Tämä siksi, että vapausasteiden liikkeiden toteuttaminen vaatii sitä. Teollisuusrobottien mekaaniset rakenteet jaetaan avoimeen kinemaattiseen rakenteeseen ja suljettuun kinemaattiseen rakenteeseen. Avoimessa kinemaattisessa rakenteessa tukivarsi kytketään edellisen perään. Se on tällä hetkellä yleisin teollisuusroboteissa käytettävä ratkaisu. Jos halutaan päästä kevyempiin rakenteisiin, voidaan tukivarret kytkeä rinnakkain. Tätä kutsutaan suljetuksi

kinemaattiseksi rakenteeksi. (Aalto ym. 1999, 15-16.)

Jokaisella robottivalmistajalla on omat ohjausjärjestelmät robottimalleilleen.

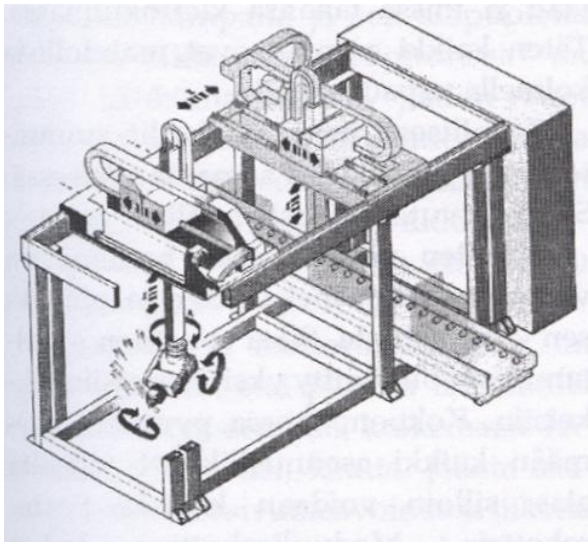
2.1.3 Robottityypit

Yleisimmät robottityypit ovat suorakulmainen robotti, sylinterirobotti, scara- robotti ja kiertyvänivelinen robotti (Aalto ym. 1999, 16).

Nimitys pääakseleiden mukaan	Rakenne	Kinemaattinen kaavio	Työalue
Suorakulmainen robotti			
Sylinterirobotti			
Napa-koordinaatistirobotti			
Scara-robotti			
Kiertyvänivelinen robotti			
Rinnakkaisrakenteinen robotti			

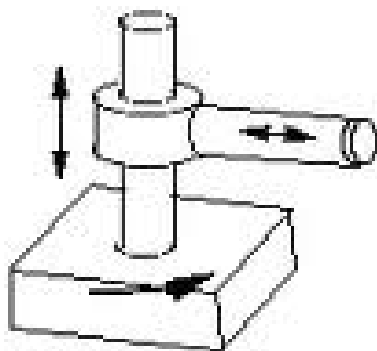
Kuva 2. Yleisimmät robottityypit (Aalto ym. 1999.)

Suorakulmainen robotti. Tyypillinen suorakulmainen robotti on portaalirobotti. (Aalto ym. 1999, 16.) Portaalirobotti toimii suoraviivaisilla liikkeillä telineessä, joka on tuettu palkeilla jokaisesta nurkastaan. Tämän rakenteen etuja ovat yksinkertaisuus ja soveltuvuus suorakulmaista liikettä tarvitseviin tehtäviin. Esimerkkejä tällaisista ovat esimerkiksi kokoonpanotyöt. (Keinänen ym. 2000, 307.) Suomen suurin robottivalmistaja Cimcorp valmistaa portaalirobotteja eri teollisuudenaloille. (Cimcorp 2010.)



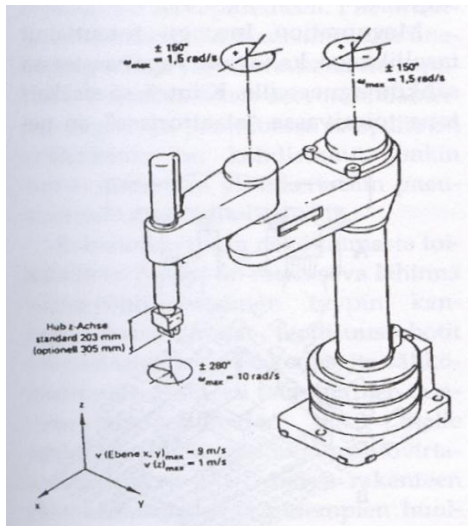
Kuva 3. Portaalirobotti (Aalto ym. 1999.)

Sylinterirobotti. Sylinterirobotin nimitys tulee sen sylinterikoordinaatistosta (Aalto ym. 1999, 17.)



Kuva 4. Sylinterirobotin rakenne (Robotics review 2011.)

Scara-robotti. Scara-robotissa robotin käsivarsi joustaa tiettyyn suuntaan. Kolmen samassa tasossa kiertyvän nivelen ja yhden lineaarisen pystyliikkeen avulla työkalu saadaan tasolla oikeaan kohtaan ja kiertymäkulmaan. Scara-robotti on ihmisen vaakatasossa liikkuvan käsivarren kaltainen, mutta sillä erolla että sen ranteeseen on asennettu pystyjohde. (Aalto ym. 1999, 16.)

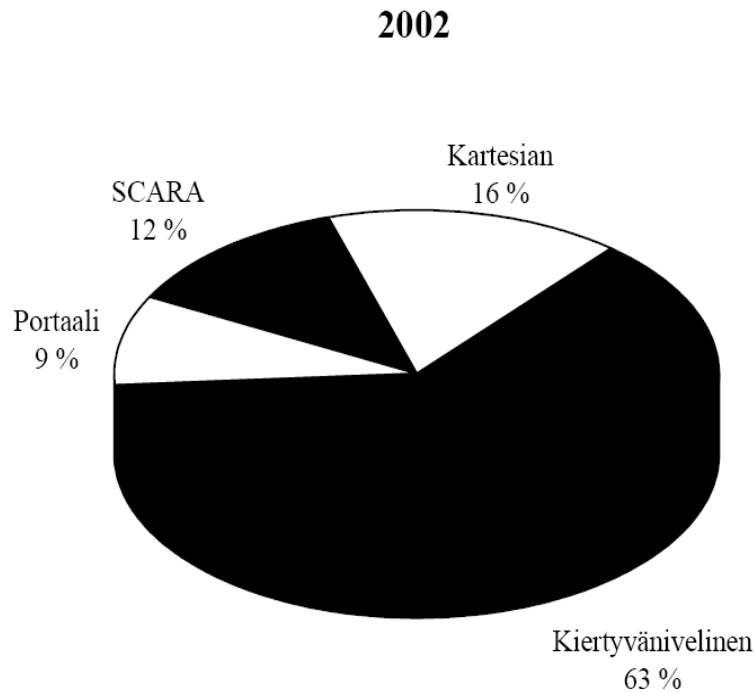


Kuva 5. Scara-robotti (Aalto ym. 1999.)

Kiertyvänivelinen robotti. Kiertyvänivelinen robotti on tyypillisin teollisuusrobotti. Siitä voidaan käyttää myös ilmaisua nivelvarsirobotti. Se pystyy robottityypeistä parhaiten matkimaan ihmiskäden liikkeitä. Nykyään kiertyvänivelisillä roboteilla on kuusi vapausastetta ja niistä vähintään kolme on kiertyviä. (Aalto ym. 1999, 16-18.)



Kuva 6. Kiertyvänivelinen robotti (ABB 2011.)



Kuva 7. Robottityyppien jakauma vuonna 2002 (Suomen Robotiikkayhdistys Ry 2002).

2.1.4 Teollisuusrobotti

Teollisuusrobottien, kuten kiertyvänivelisen robotin, ohjausjärjestelmä perustuu servotekniikkaan. Sen suurena etuna on tarkkuus, minkä vuoksi se soveltuu erinomaisesti teollisuusrobotteihin. Servojärjestelmä sisältää moottorin, aseman ja nopeuden mittauslaitteet sekä ohjausjärjestelmän. Lisäksi servojärjestelmiin sisältyy liikkeenohjausjärjestelmiä, jotka takaavat robotin tarkan ohjauksen. (Keinänen ym. 2007, 157.)

Robotin tukivarsien liikuttamiseen käytetään servomoottoreita. Ne ovat joko tasavirtaservomoottoreita tai hiiliharjattomia vaihtovirtaservomoottoreita. Jälkimmäisen etuina ovat rakenteen yksinkertaisuus ja pidemmät huoltovälit. (Aalto ym. 1999, 19.)



Kuva 8. AC- ja DC-servomootorit (Aseko 2010.)

Robotin tehokas toiminta edellyttää tietoja sen sijainnista ja nopeudesta. Näiden tietojen keräämiseksi tarvitaan antureita, jotka aistivat jatkuvasti robotin tilaa ja tarvittaessa rajoittavat robotin liikkumista. Anturit voidaan jakaa analogisiin, digitaalisiin ja rajakytkimiin. (Keinänen ym. 2000, 318-319.)

Kiertyvänivelisten robottien servomoottoreiden aseman mittaukseen on perinteisesti käytetty resolveria, joka kuuluu analogisiin antureihin. Resolveri on induktiivinen liikeanturi, josta käytetään myös nimitystä differentiaalimuuntaja. (Keinänen ym. 2007, 202.) Se on kiinnitettynä jokaiseen niveltä liikuttavaan servomoottoriin. Moottorin akselin kiertymiskulmaa seuraamalla resolveri saa tietoja robotin asemapaikasta ja välittää tämän tiedon ohjausjärjestelmälle. (Aalto ym. 1999, 30-31.)

Nykyään aseman ja nopeuden mittaaminen voidaan tehdä samalla anturilla. Mittaamiseen voidaan käyttää esimerkiksi pulssi- eli inkrementtianturia. (Aalto ym. 1999, 31.) Tällöin asema saadaan selville pulsseja laskemalla ja nopeus pulssien taajuuden avulla. (Keinänen ym. 2000, 191.)



Kuva 9. Pulssiantureita (Penlink 2010.)

Ohjausjärjestelmällä tarkoitetaan yhden koneen tai toiminnon ohjausta. Tämänkaltaisen ohjausjärjestelmä voi olla esimerkiksi ohjelmoitava logiikka tai mikrotietokone. (Keinänen ym. 2007, 211.) Ohjelmoitavan logiikan tuloportteihin kytketään anturit, jotka keräävät tietoa prosessiin tilasta. Lähtöportteihin kytketään prosessiin kuuluvat toimilaitteet, jotka suorittavat robotin liikkeet ja tehtävät. Toimilaitteita ovat esimerkiksi sähkömoottorit ja releet. Logiikka ohjaa toimilaitteita antureilta saatujen tietojen ja sen omaan muistiin ohjelmoidun ohjelman avulla. (Keinänen ym. 2000, 243.) Tällainen ohjausjärjestelmä pystyy toimimaan itsenäisesti, mutta yleensä se on osana kokonaista automaatiojärjestelmää (Keinänen ym. 2007, 210).

Teollisuusrobotin tarkkaan paikoittamiseen tarvitaan erityinen liikkeenohjausjärjestelmä. Robotin monimutkaisuuden vuoksi robottivalmistajat valmistavat yleensä omat liikkeenohjausjärjestelmät. Liikkeenohjain paikoittaa robotin tarkasti haluttuun kohtaan robottiin tehdyn ohjelman mukaisesti. Paikoitustehtävä voi tulla myös ylemmän tason ohjaukselta. (Keinänen ym. 2007, 214.)



Kuva 10. AC-servomootorin ohjain (Kiowa 2010.)

2.1.5 Robotin työkalut

Robotin työkalut voidaan jakaa kahteen eri ryhmään: tarraimet, työkaluilla varustetut tarraimet ja varsinaiset työkalut. (Keinänen ym. 2000, 321).

Valittaessa tarrainta täytyy ottaa huomioon kaikki mahdolliset tarraintyyppit ja tartuntatavat, työstettävän kappaleen koko, muodot ja käytettävä materiaali. Myös robotin oma toimintaympäristö asettaa omat rajoituksensa. Tarttujiin löytyy vakioratkaisuja, mutta usein valikoima on liian suppea. (Keinänen ym. 2000, 319-320.) Usein tarrainta muokkaamalla päästään haluttuun ratkaisuun, mutta joskus tarttuja joudutaan suunnittelemaan itse. Suunnittelu vaatii hyvin tarkkoja laskelmia ja hyviä resursseja. Tärkeitä seikkoja suunnittelussa ovat esimerkiksi se, vaatiiko kappaleen työstö kääntelyä, vaihtelevatko koko, muoto ja massa toistuvasti. Taloudelliselta kannalta täytyy miettiä, millä tavoin suunnittelussa voitaisiin päästä mahdollisimman edulliseen, mutta kuitenkin toimivaan ratkaisuun. (Aalto ym. 1999, 65-66.)

Tarrainta suunniteltaessa ja valittaessa on oltava tiedossa eri tarraintyyppit ja tartuntatavat. Tarraimet jaetaan seuraaviin ryhmiin:

- avautuvat ja sulkeutuvat tarraimet; kappaleisiin tartutaan joko ulko- tai sisäpuolisella otteella
- kiertyväsormiset ja rinnakkain suoraviivaisesti liikkuvilla sormilla varustetut tarraimet
- pneumaattiset, hydrauliset tai sähköiset tarraimet
- liikkuvat kaksi-, kolmi- ja useampisormiset tarraimet,
- jäykät ja joustavat tarraimet
- kappalekohtaiset tai yleistarraimet sen määrätyn kappalemäärän mukaan, johon tarraimella voidaan tarttua

- keskittävät tarraimet, jotka siirtävät kappaletta otetta muodostettaessaan vakioasemaan
- magneettisesti toimivat tarraimet
- alipaineella toimivat tarraimet
- yksittäinen, kaksois- tai revolveritarrain (monitarrain) sen mukaan, kuinka monta sormiryhmää tarraimessa on
- älykkäät anturoidut tarraimet,
- erikoistarraimet. (Aalto ym. 1999, 60.)

2.1.6 Robotin koordinaatistot

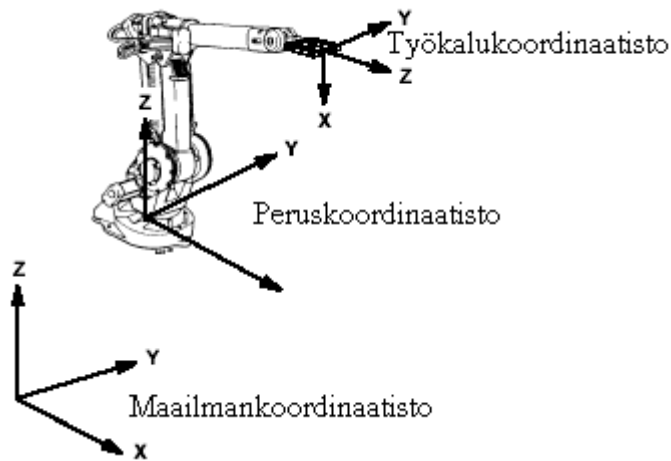
Teollisuusrobotti liikkuu XYZ-suunnissa määriteltyjen numeroarvojen eli koordinaattien mukaan. Robotin liikkeet tapahtuvat useiden koordinaatistojen avulla. Nämä koordinaatistot on määritelty suhteessa toisiinsa. Yleisesti tunnetut koordinaatistot ovat:

- Maailmankoordinaatisto
- Peruskoordinaatisto
- Työkalukoordinaatisto. (Keinänen ym. 2007, 260.)

Maailmankoordinaatisto on teollisuusrobotin työympäristöön sidottu koordinaatisto. Työympäristö voi käsittää esimerkiksi rakennuksen tai oheislaitteet. (Keinänen ym. 2007, 260.)

Peruskoordinaatisto on robotin jalustaan sidottu koordinaatisto. XY-suuntainen vaakasuorataso muodostuu robotin jalustan tasolle. Z-akseli on pystysuorainen akseli, joka kulkee ensimmäisen nivelen keskikohdan kautta. (Keinänen ym. 2007, 261.)

Työkalukoordinaatisto on sidottu työkalun tai käsiteltävän kappaleen koordinaatistiksi (Keinänen ym. 2007, 261).



Kuva 11. Robotin koordinaatistot (CADblog 2011.)

2.2 Robotin ohjelmointi

Robottien ohjelmointitavat jakaantuvat kahteen pääluokkaan, jotka ovat online-ohjelmointi ja offline-ohjelmointi.

2.2.1 Online-ohjelmointi

Online-ohjelmointitavassa robottia tarvitaan ohjelmoinnissa, joten se ei voi olla tuotannossa mukana. Online-ohjelmointitapoja ovat opettamalla ohjelmointi, käsiohjaimella ohjelmointi, näyttämällä ohjelmointi ja konenäköpohjainen ohjelmointi. (Malm 2008, 95.)

Opettamalla ohjelmointi. Robotin ohjelmointiin käytetään käsiohjainta. Ohjaimella robotti ajetaan haluttuun pisteeseen, joka tallennetaan muistiin. Sitten siirrytään seuraavaan pisteeseen. (Malm 2008, 95.)

Käsiohjaimella ohjelmointi. Oliopohjaisessa ohjelmoinnissa käytetään valmiita ikoneita ohjelman teossa. Ohjelmointi voi olla myös valikkopohjaista tai tekstipohjaista, joka tosin vaatii ohjelmointikielen hallintaa. (Malm 2008, 95.)



Kuva 12. Robotin käsiohjain (RobotWorx 2011).

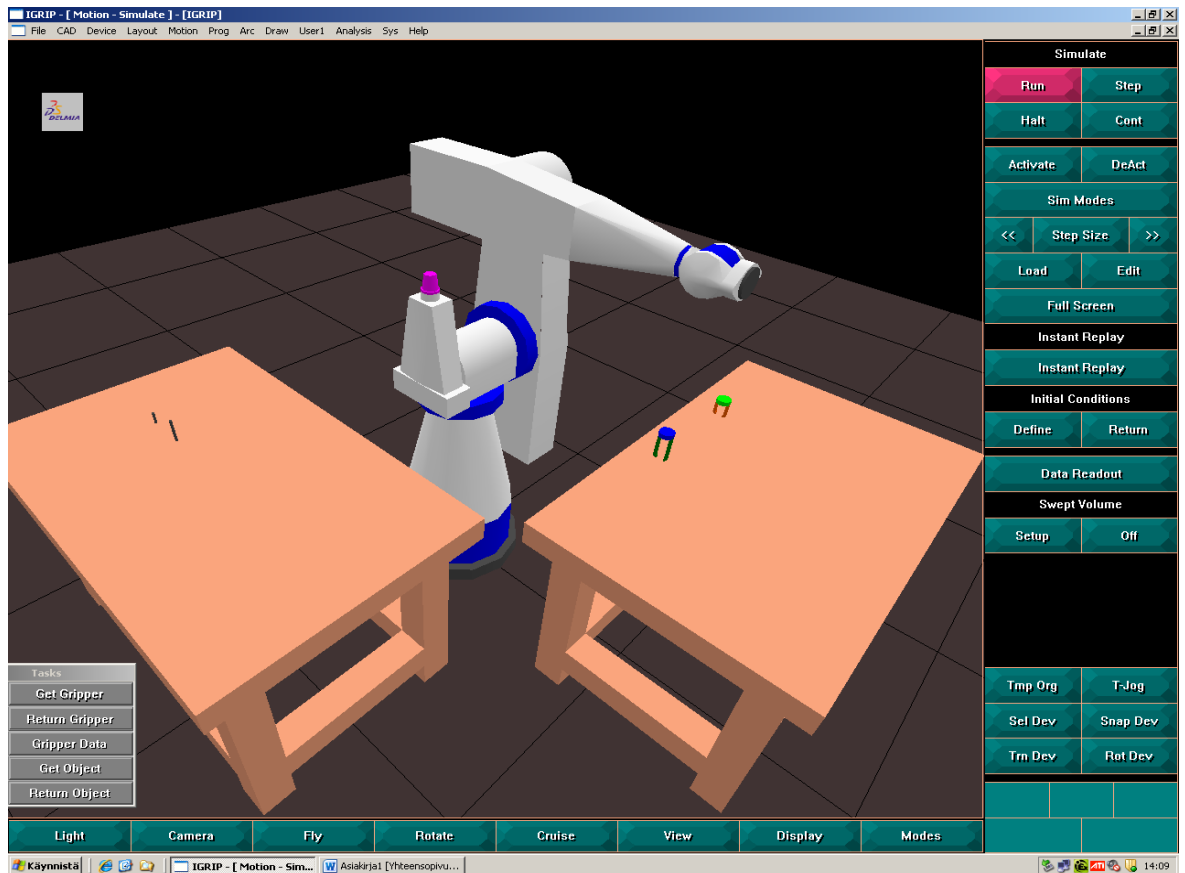
Näyttämällä ohjelmointi. Robotille näytetään taluttamalla haluttu asema tai opetetaan haluttu liikerata (Malm 2008, 96). Maalausrobotit ovat yleisimpiä näyttämällä ohjelmoituja robotteja. Tämän ohjelmointitavan heikkoutena on liikeradan epätarkkuus. (Aalto ym. 1999, 78.)

Konenäköpohjainen ohjelmointi. Aluksi kohteesta otetaan kuva, jonka perustella konenäköjärjestelmä päättelee itsenäisesti pisteensä. Riippuen järjestelmän itsenäisyydestä käyttäjä hyväksyy tai hylkää liikeradat ja makrot. (Malm 2008, 96.)

2.2.1 Offline-ohjelmointi

Robottien offline-ohjelmoinnilla eli etäohjelmoinnilla tarkoitetaan robotin uudelleenohjelmointia järjestelmän ulkopuolisessa tietokoneessa. Se on helpottanut ja nopeuttanut suunnittelua huomattavasti. Etäohjelmoinnin avulla ohjelmointi voidaan suorittaa ilman, että robotin toimintaa tarvitsee keskeyttää. Simuloimalla voidaan kokeilla ja testata erilaisia ratkaisuja, ennen kuin ne otetaan käyttöön esimerkiksi tehdasympäristössä. Tällä tavalla voidaan välttää suunnitteluvaiheen virheitä ja arvioida muutosten vaikutusta kokonaisuuteen. Robottijärjestelmää simuloitaessa voidaan kokeilla erilaisia layout-ratkaisuja, määrittää robotin työkierto ja ulottuvuudet, arvioida työhön kuluva aika sekä testata mahdollisia vaaratilanteita. Simulointia voidaan hyödyntää myös markkinoinnissa ja uusien työntekijöiden koulutuksessa. (Aalto, & Lylynoja, 2011.)

Offline-ohjelmoinnissa ja simuloinnissa käytettäviä tietokoneohjelmistoja ovat esimerkiksi Delmian IGRIP (kuva 13) ja ABB:n RobotStudio.



Kuva 13. IGRIP-ohjelmointi ja simulointiohjelmisto.

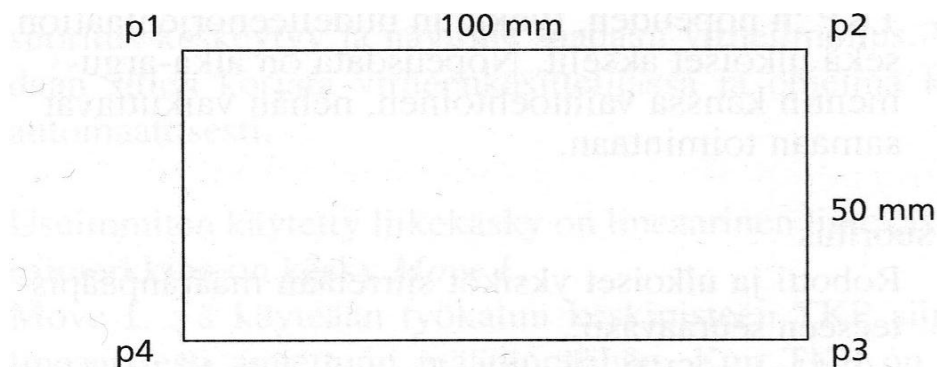
Offline-ohjelmointitapoja ovat tekstipohjainen etäohjelmointi, etäohjelmointi virtuaalisella käsiohjaimella ja mallipohjainen etäohjelmointi (Malm 2008, 95).

Tekstipohjainen etäohjelmointi. Ohjelmointi tehdään tietokonetta käyttäen. Tämä tapa mahdollistaa monimutkaiset ohjelmarakenteet, mutta vaatii ohjelmointikielen hyvää hallitsemista. (Malm 2008, 97.)

ABB:n roboteissa käytetään käskyjen ohjelmointiin RAPID-ohjelmointikieltä. RAPID-kieli koostuu päärutiinista (pääohjelma), alirutiineista (aliohjelmista) ja ohjelmadatasta. Päärutiini aloittaa aina ohjelman ja kutsuu alirutiinien toimintoja käyttöön. Käskeysarja sisältää toiminnon, jonka robotin halutaan tekevän, esimerkiksi liikkuminen paikasta toiseen. Käskyjä noudatetaan siinä järjestyksessä

kuin ne on ohjelmoitu. (Keinänen ym. 2007, 263)

Liikekäslynä käytetään usein MoveL-komentoa. MoveL-komennolla pystytään siirtämään robotin työkalu haluttuun sijaintiin. Esimerkkinä on ohjelma, jossa työkalu aloittaa pisteestä p1 ja sen jälkeen annetaan koordinaatit uuteen pisteeseen. Kun työkalu on kiertänyt kaikki pisteet, se palaa pisteeseen p1. (Keinänen ym. 2007, 264.)



MoveL p1....

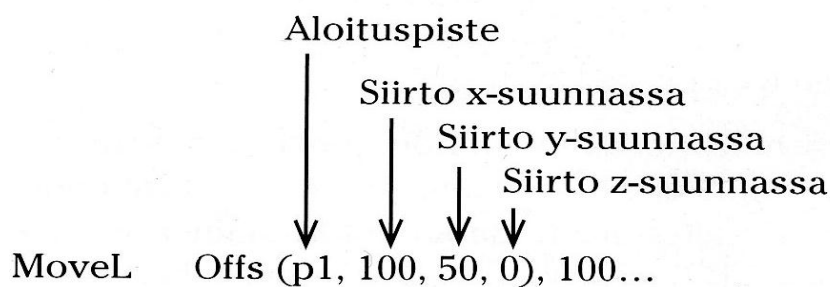
MoveL p2 MoveL offs (p1,100,0,0)...

MoveL p3 MoveL offs (p1,100,50,0)...

MoveL p4 MoveL offs (p1,0,50,0)...

Kuva 14. RAPID-ohjelmointiesimerkki (Keinänen ym. 2007.)

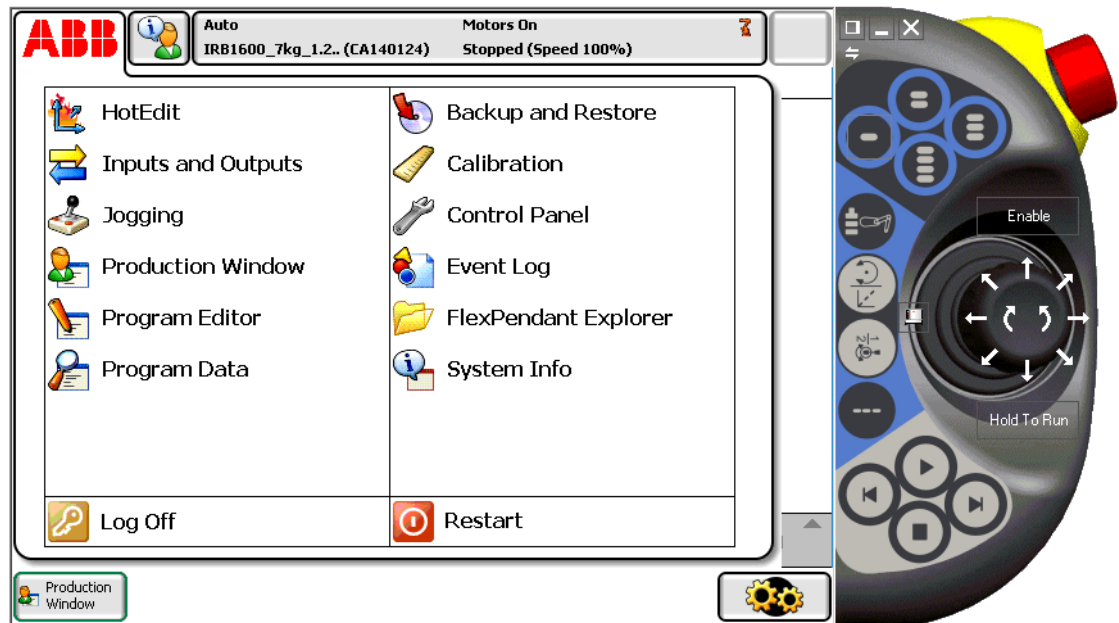
Uuden pisteen koordinaatit annetaan suhteessa pisteeseen p1. Tämä tapahtuu offsettina eli siirtona edellisestä pisteestä seuraavaan. Liikekäslynä käytetään komentoa MoveL. Ympyräliikkeissä käytetään komentoa MoveC ja MoveJ-komento liikuttaa robottia sen nivelistä. (Keinänen ym. 2007, 264-265.)



Kuva 15. Koordinaattien selitykset (Keinänen ym. 2007.)

Etäohjelmointi virtuaalisella käsiohjaimella. Ohjelmointi tapahtuu tietokoneella käsiohjainta emuloivalla ohjelmalla. (Malm 2008, 98.)

Esimerkkinä ABB:n RobotStudio-ohjelmasta löytyvä Virtual FlexPendant -toiminto, joka simuloi ABB:n robotin ohjainta (kuva 16). Virtual FlexPendantissa on samat ominaisuudet kuin ABB:n robottiohjaimessa. (ABB [viitattu 11.4.2011].)



Kuva 16. Virtual FlexPendant.

Mallipohjainen etäohjelmointi. Ohjelmointi tehdään ilman robottia tietokoneella 3D-graafisessa käyttöliittymässä. Valmis ohjelma testataan ensin simuloimalla. (Malm 2008, 98.)

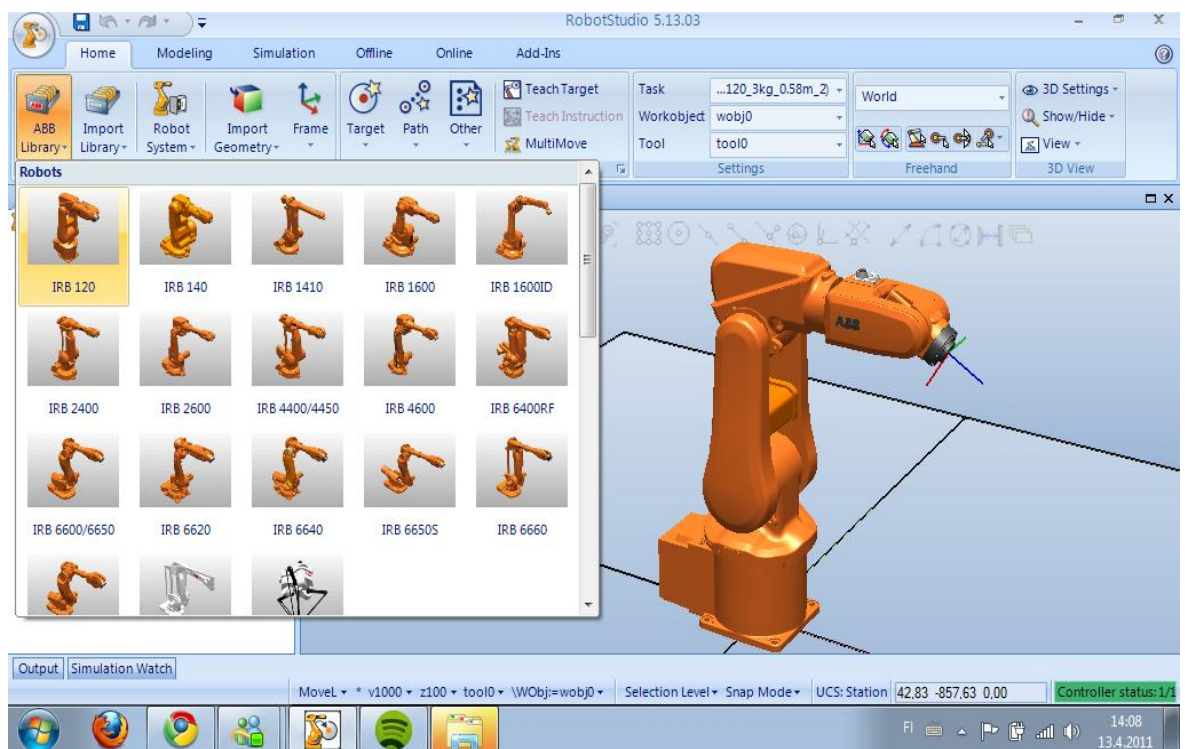
Taulukko 2. Ohjelmointimenetelmät (Malm 2008).

ON-LINE	OFF-LINE
Opettamalla ohjelmointi	Tekstipohjainen etäohjelmointi
Oliopohjainen ohjelmointi	Oliopohjainen ohjelmointi
Tekstipohjainen ohjelmointi	”Etäohjaimella” ohjelmointi
Näyttämällä ohjelmointi	Graafinen etäohjelmointi
Konenäköpohjainen ohjelmointi	Automaattinen etäohjelmointi

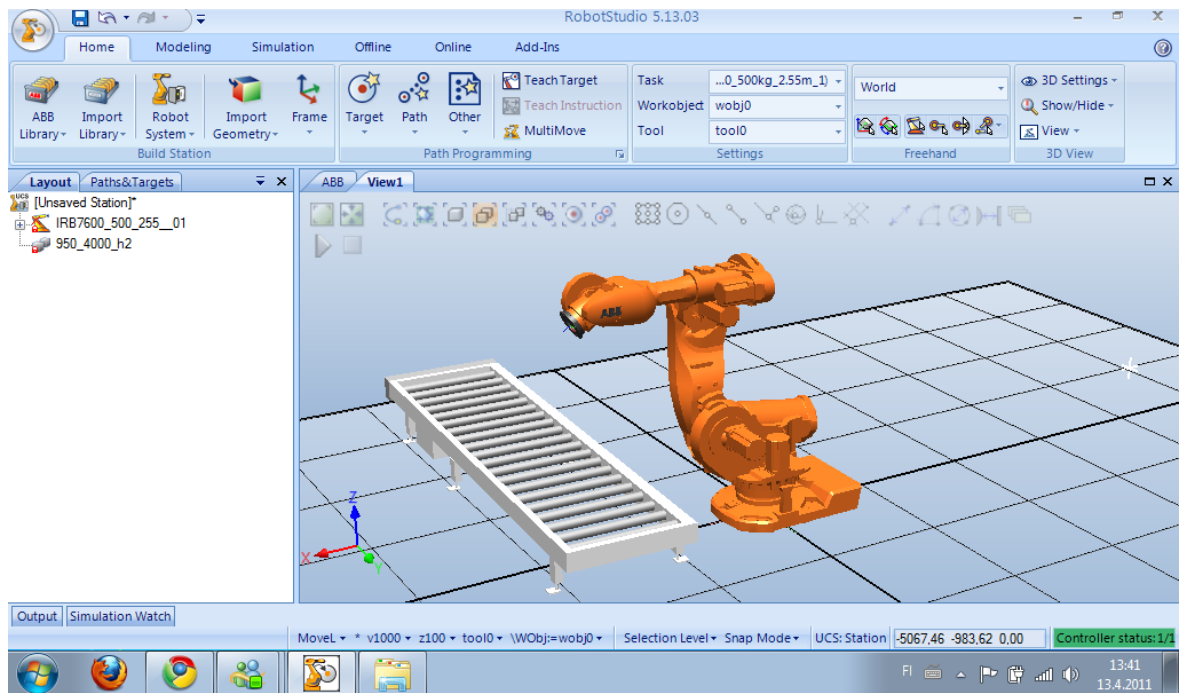
2.3 ABB RobotStudio-ohjelma

RobotStudio on ABB:n kehittämä mallinnus- ja etäohjelmointiohjelmisto omille robottimalleilleen. Ohjelmassa käytettävä ohjelmointikieli robottiohjelmien tekemiseksi on täysin sama kuin ABB:n roboteissa. Tämä tuo sen edun, että RobotStudiolla tehtyjä ohjelmia voidaan hyödyntää sellaisenaan myös todellisessa järjestelmässä. Offline-ohjelmoinnin avulla robotti voidaan ohjelmoida etukäteen ja tuotantoa keskeyttämättä. (ABB [viitattu 11.4.2011].)

RobotStudio-ohjelman kirjastoista löytyvät robottimallien (kuva 17) lisäksi valmiiksi mallinnettuja työkaluja ja oheislaitteita, kuten esimerkiksi kuljettimia (kuva 18) Itse luotuja CAD-kuvia voidaan tuoda ohjelmaan yleisimmissä tiedostomuodoissa, joita ovat esimerkiksi ACIS, IGES, STEP, VDAFS. (ABB [viitattu 11.4.2011].)



Kuva 17. RobotStudio-ohjelman ABB-library ja Robottimalli IRB 120



Kuva 18. Robottimalli IRB7600 ja rullakuljetin

3 TYÖN TOTEUTUS

3.1 Kuvan muokkaus

Aiheen varmistuttua suunnittelu aloitettiin tutkimalla RobotStudio-ohjelman resursseja. Tilasta, jossa robotti sijaitsee, oli olemassa valmis CAD-ohjelmalla simuloitu kuva. Koko tilaa ei ollut tarkoitus liittää RobotStudioon, vaan ainoastaan ABB-robotin solu. Kuvaa kokeiltiin alustavasti kokonaisuutena liittää ohjelmaan, mutta tehtävä epäonnistui. Ongelmana oli se, että alkuperäisessä 3D-mallissa pintojen määrä oli liian suuri. Myös tiedostoformaatti ei ollut sopiva RobotStudio-ohjelmaan. Asiaa tiedusteltaessa ABB:ltä selvisi, että perustiedostomuoto, jota RobotStudio tukee on ACIS (.sat- tiedostopääte). Muut tiedostomuodot ovat IGES, STEP ja VDAFS.

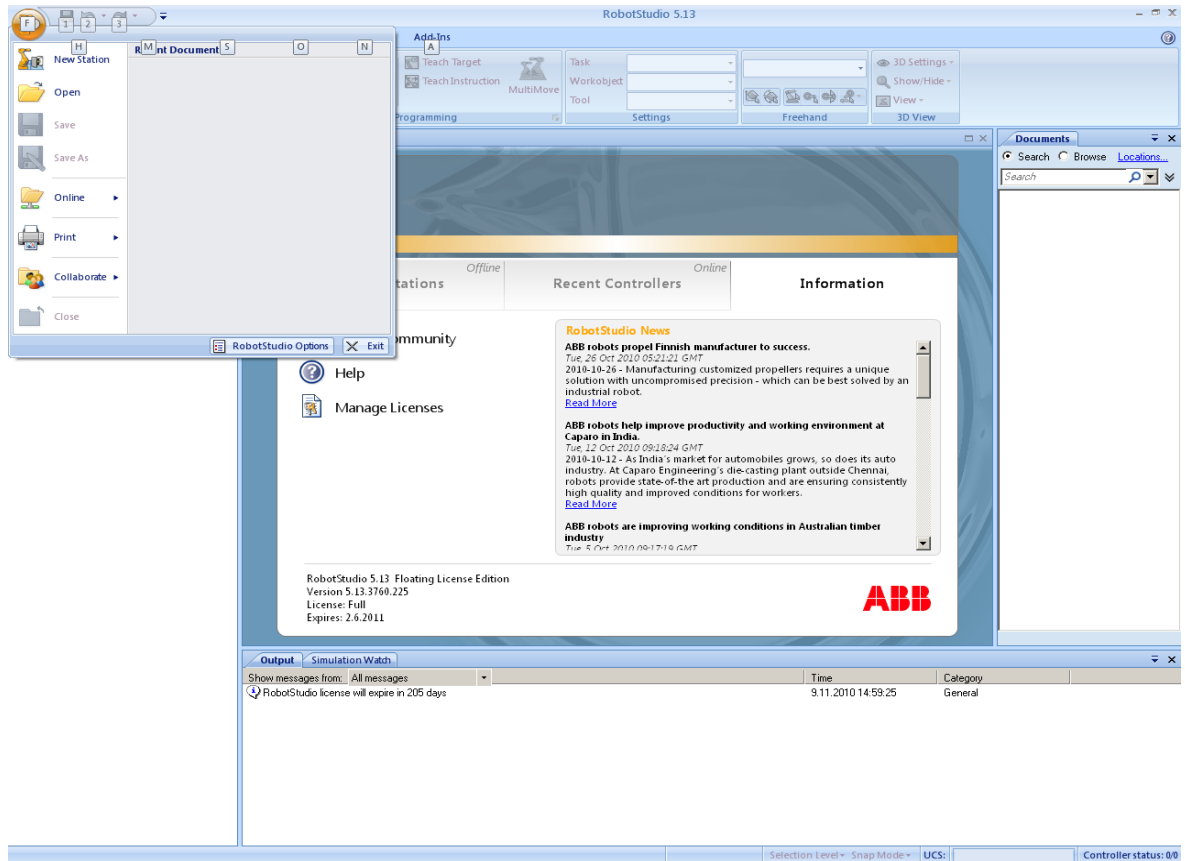
Ongelma kyseisten tiedostomuotojen kanssa on se, että ne eivät tue meshiä. Tämä aiheuttaa sen, että jokaisen pinnan 3D-koordinaatit määritellään moneen kertaan, mikä nosti tiedostokoot isoiksi. Meshillä tarkoitetaan 3D-verkkomallia. Siinä polygonipinnat ovat indeksoituja ja määritellään vain kerran. (Hellman 2006.)

Seuraavaksi kuvasta karsittiin turhia pintoja Rhinoceros-ohjelmalla. Käytännössä tämä tarkoitti sitä, että ainoastaan ABB-robotin solu jätettiin paikalleen. Tästä versiosta käännettiin useita eri versioita RobotStudio-ohjelmaa varten. ACIS-tiedostomuotoon käännetty versio oli lopulta ainoa toimiva. Muiden tiedostomuotojen, kuten IGES tai STEP, käyttö ei ollut mahdollista lisessipuutteiden vuoksi. Sen jälkeen kun toimiva versio löytyi, solun siirtämistä RobotStudio-ohjelmaan voitiin kokeilla.

3.2 RobotStudio -ohjelman käynnistys

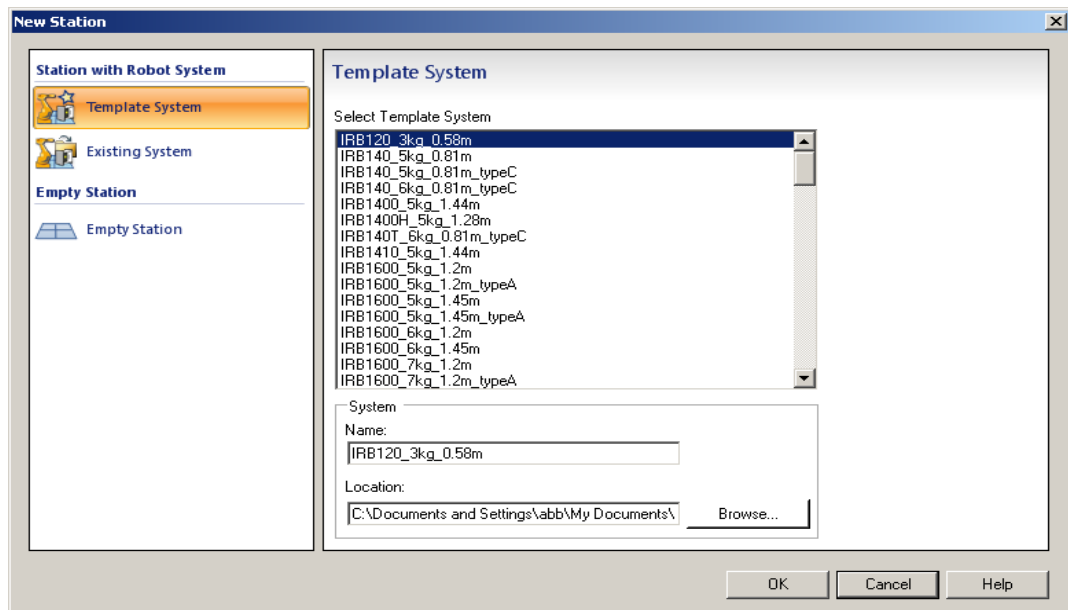
RobotStudio-ohjelma voidaan käynnistetään joko Windows-käyttöjärjestelmän Käynnistä-valikon kautta tai mahdollisesta työpöydällä olevasta kuvakkeesta. Ohjelman aloitusvalikossa (kuva 19) voidaan valita aloitetaanko kokonaan uusi

asema vai avataanko aiemmin aloitettu asema. Sama valinta voidaan suorittaa ohjelman vasemmassa ylänurkassa olevasta RobotStudioButton-painikkeesta. Saman painikkeen kautta asema voidaan myös tallentaa .



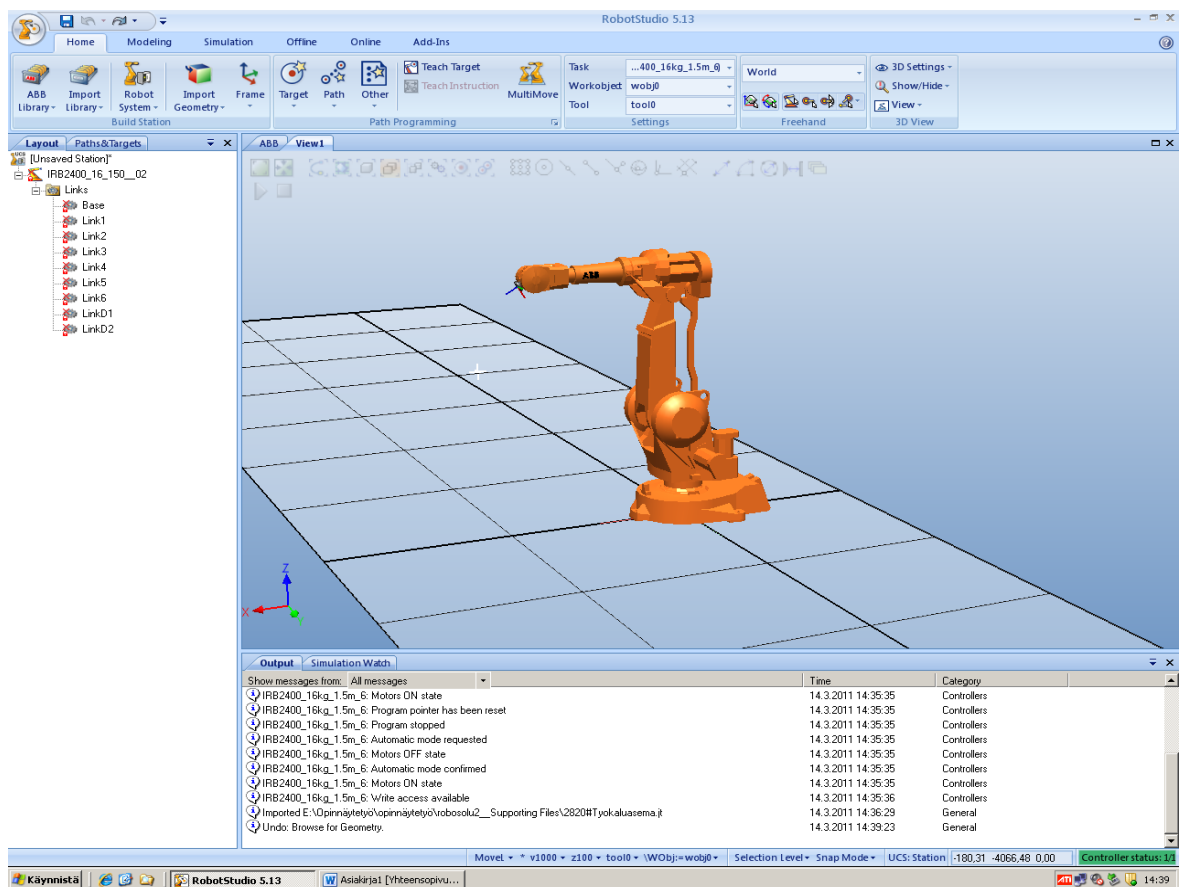
Kuva 19. RobotStudion alkuvalikko (ABB 2010).

Uuden aseman valinnan jälkeen valitaan robottimalli Template System -välilehden alta (kuva 20). Seinäjoen ammattikorkeakoulun, Tekniikan yksikön, robottilaboratoriosta löytyvä robottimalli on IRB2400_16kg_1.5m, joten tähän työhön valittiin tämä kyseinen malli.



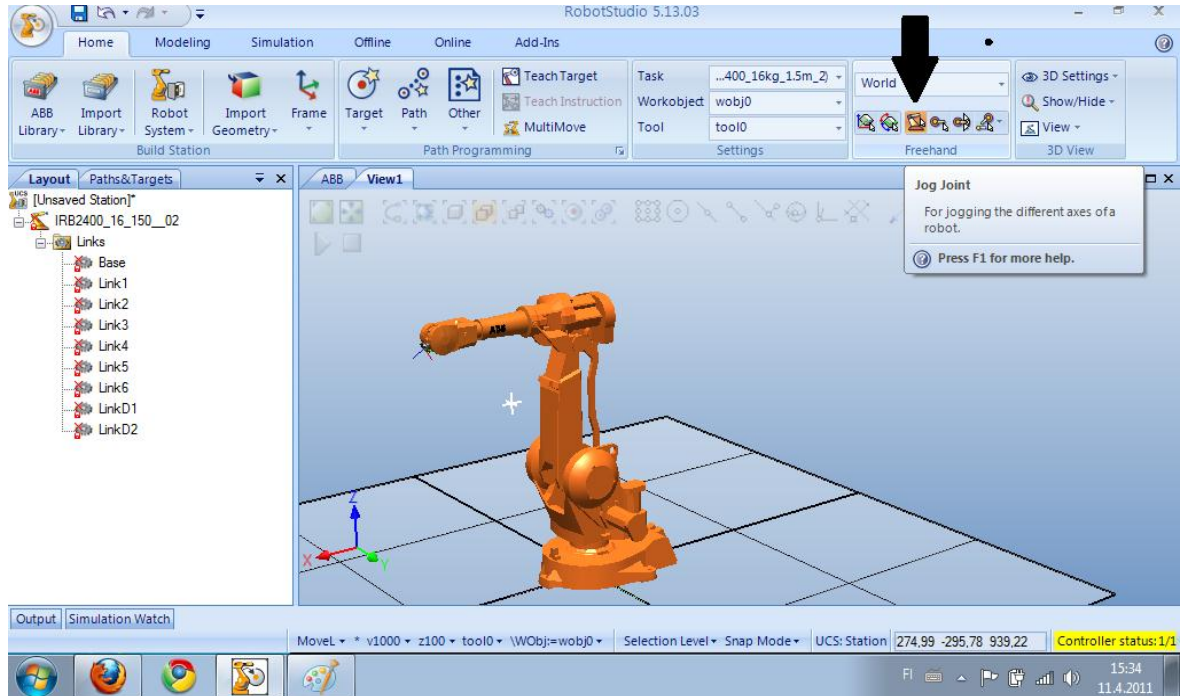
Kuva 20. Robottimallin valitseminen (ABB 2010).

Robottimallin valinnan jälkeen asema avautui valitulla mallilla (kuva 21).

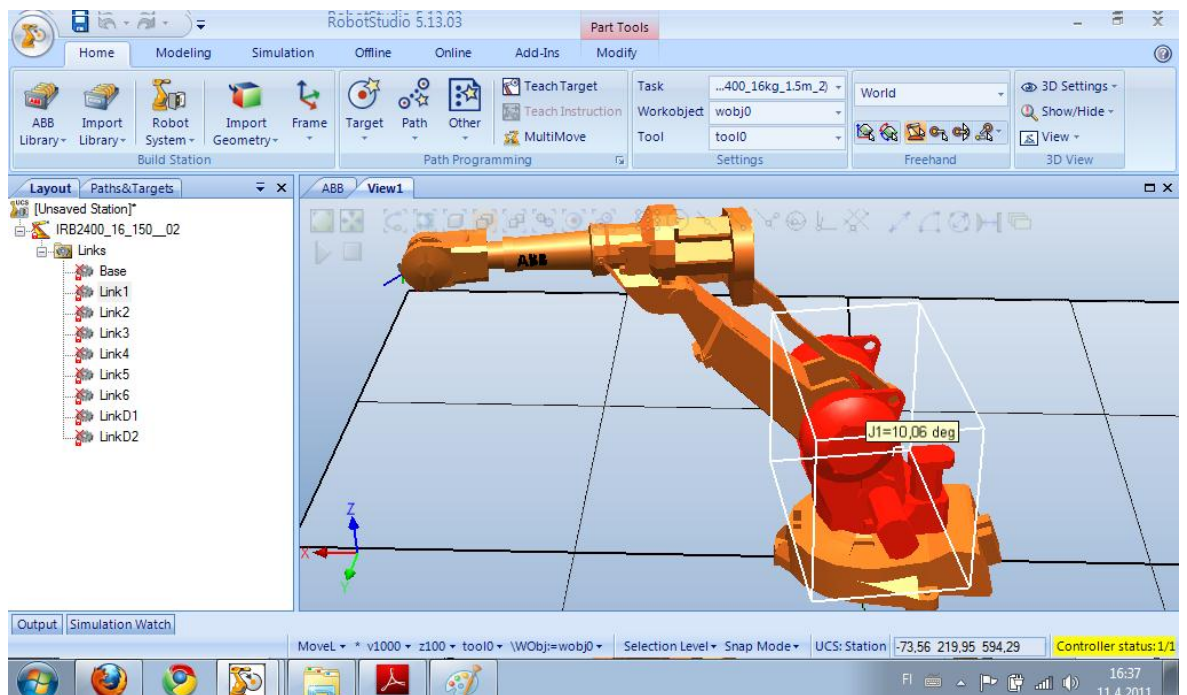


Kuva 21. Robottimalli IRB2400

Robottia voidaan liikuttaa sen nivelistä Jog Joint -toiminnolla (kuva 22). Nivelä liikutetaan valitsemalla haluttu nivel ja pitämällä hiiren vasen näppäintä pohjassa. Nivel muuttuu punaiseksi ja liikkuu samalla kun hiirtä liikutetaan (kuva 23).

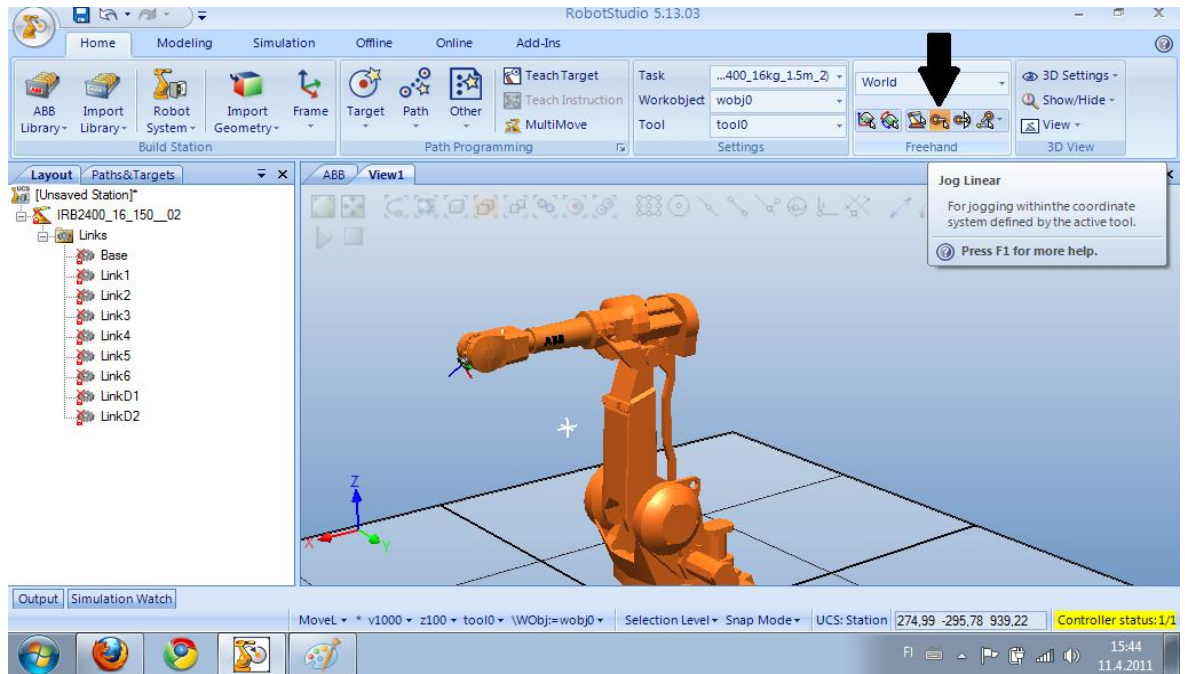


Kuva 22. Jog Joint -toiminto.

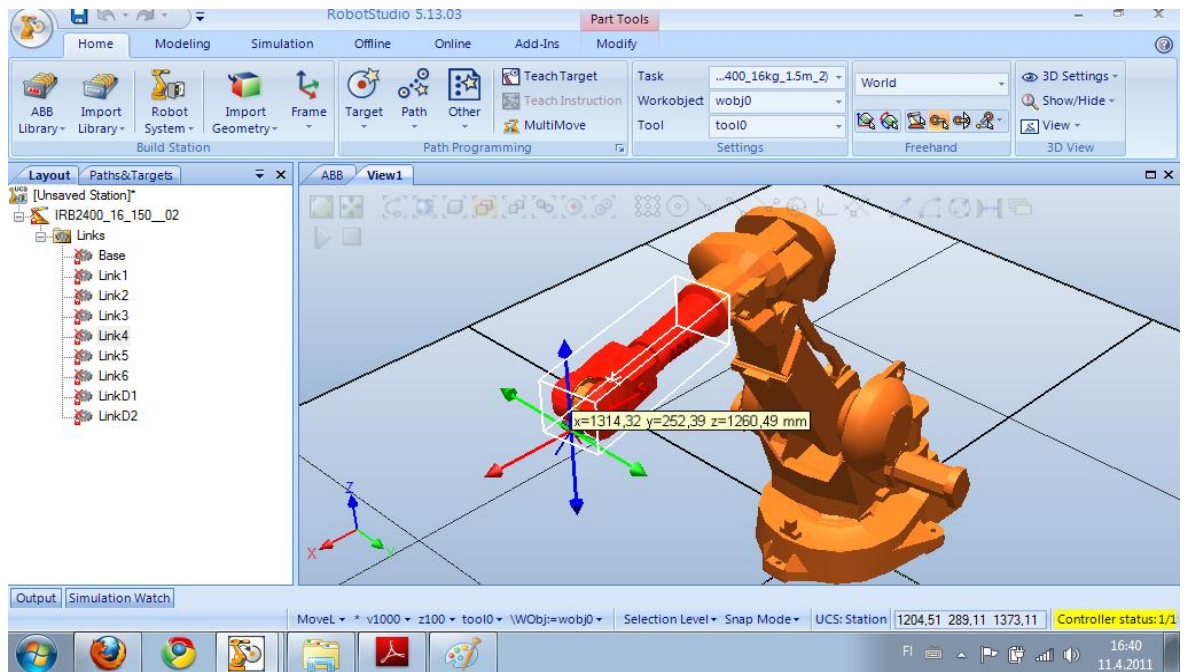


Kuva 23. Robotin nivelten liikuttaminen Jog Joint -toiminnolla

Jog Linear -toiminnolla (kuva 24) robottia voidaan liikuttaa maailmankoordinaatiston mukaisesti. Kun valittu nivel muuttuu punaiseksi, sitä voidaan liikuttaa viereen ilmestyvän koordinaatiston mukaisesti (kuva 25).



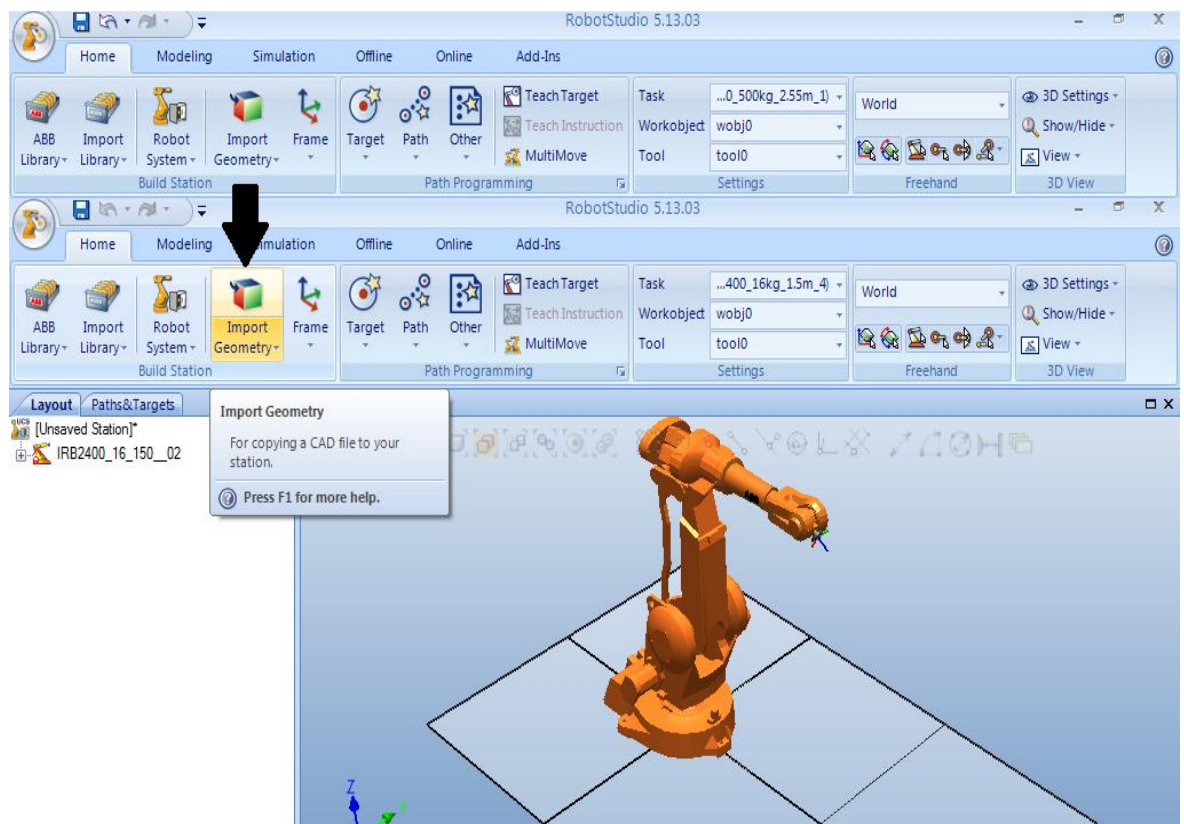
Kuva 24. Jog Linear -toiminto.



Kuva 25. Robotin nivelten liikuttaminen Jog Linearilla -toiminnolla.

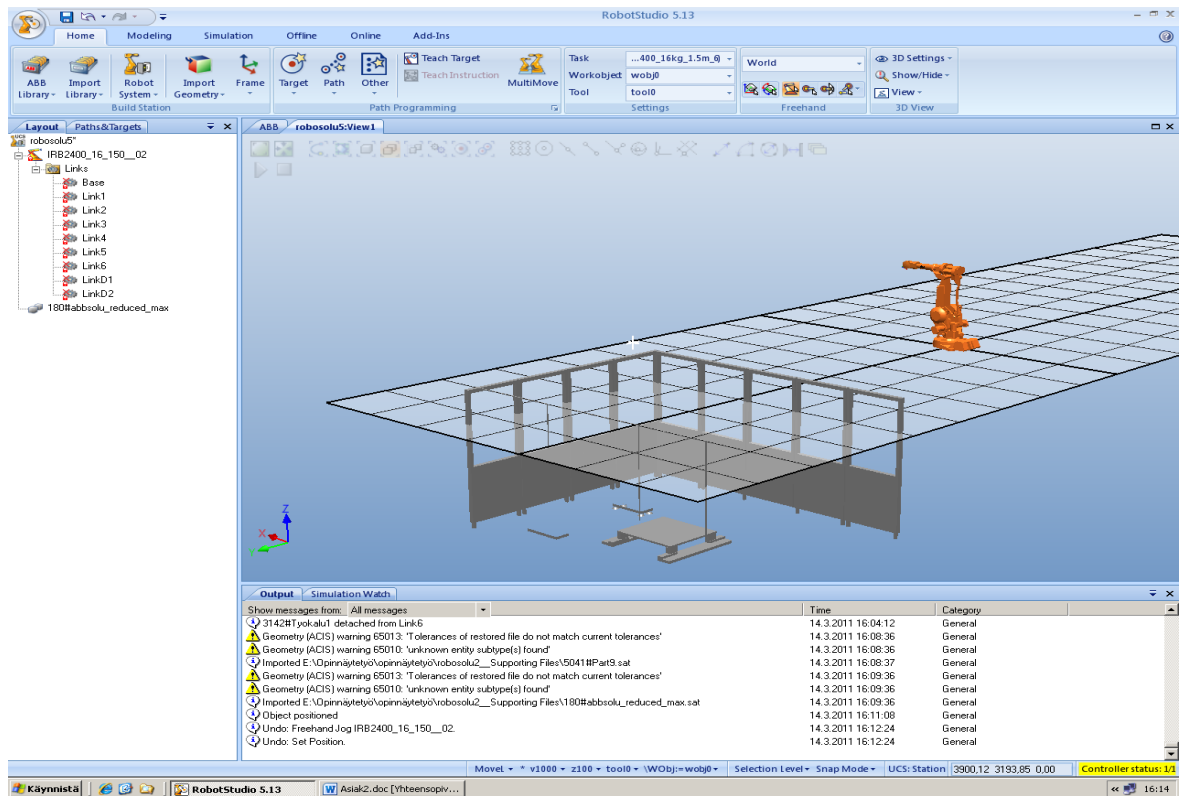
3.3 Robottisolun siirto ohjelmaan

RobotStudio-ohjelmaan voidaan tuoda import geometry -toiminnon avulla itse cad-ohjelmalla luotuja tiedostoja. ACIS-tiedostomuotoon (.sat -pääte) käännettyt solun osat voidaan siis tuoda tämän toiminnon kautta ohjelmaan. Import geometry-toiminto löytyy Home-välilehdeltä neljäntenä vasemmalta (kuva 26). Hiiren vasenta näppäintä painamalla avautuvasta valikosta valitaan Browse for Geometry.



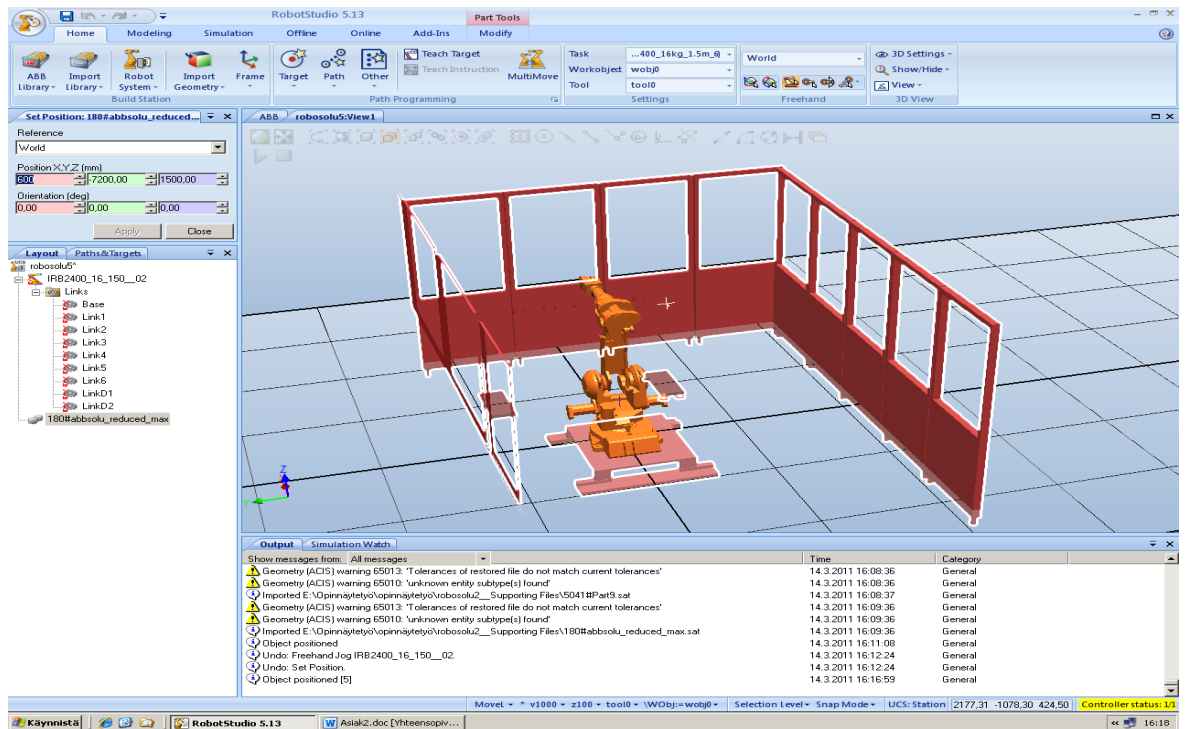
Kuva 26. Import Geometry -toiminto

Aluksi tuotiin Abbsolu_reduced_max -tiedosto, joka sisältää seinät ja robotin jalustan (kuva 27). Avatut geometriatiedostot tulivat näkyviin Layout-valikkoon ruudun vasemmalle puolelle.



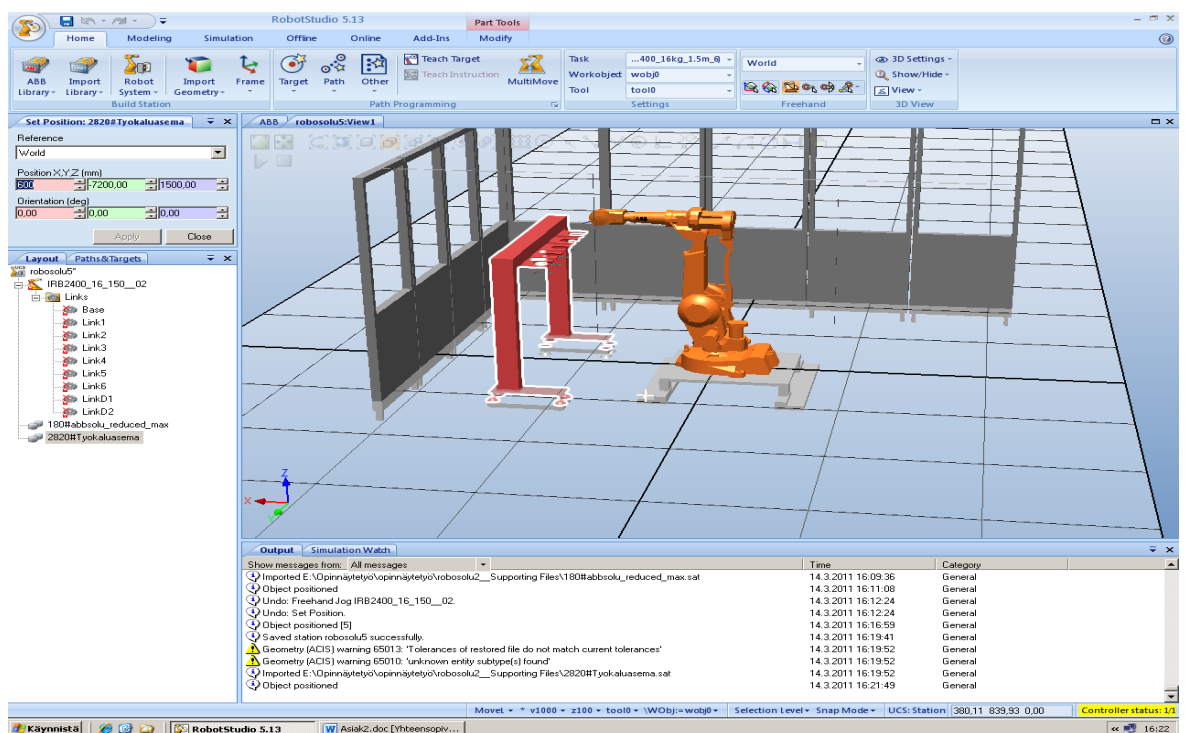
Kuva 27. Robotin jalustan ja solun liittäminen.

Kuvasta 27 nähdään, että seinät ja jalusta eivät ole oikeassa paikassa robottiin nähden. Seinien ja alustan koordinaattien numeroarvoja pystyttiin muuttamaan set position -tilan kautta (kuva 28). Hiiren vasenta näppäintä painamalla valittiin haluttu kappale, joka muuttui punaiseksi. Tämän jälkeen painettiin hiiren oikeaa näppäintä, jonka jälkeen avautuvasta valikosta valittiin set position -toiminto. Sen jälkeen näytön vasempaan reunaan avautui set position -valikko. Tästä valikosta aktiivisen kappaleen sijaintia pystyttiin siirtelemään vaihtamalla numeroarvoja xyz-suunnissa (Position x,y,z (mm)) ja kappaleen kulmaa asteissa (Orientation (deg.)).



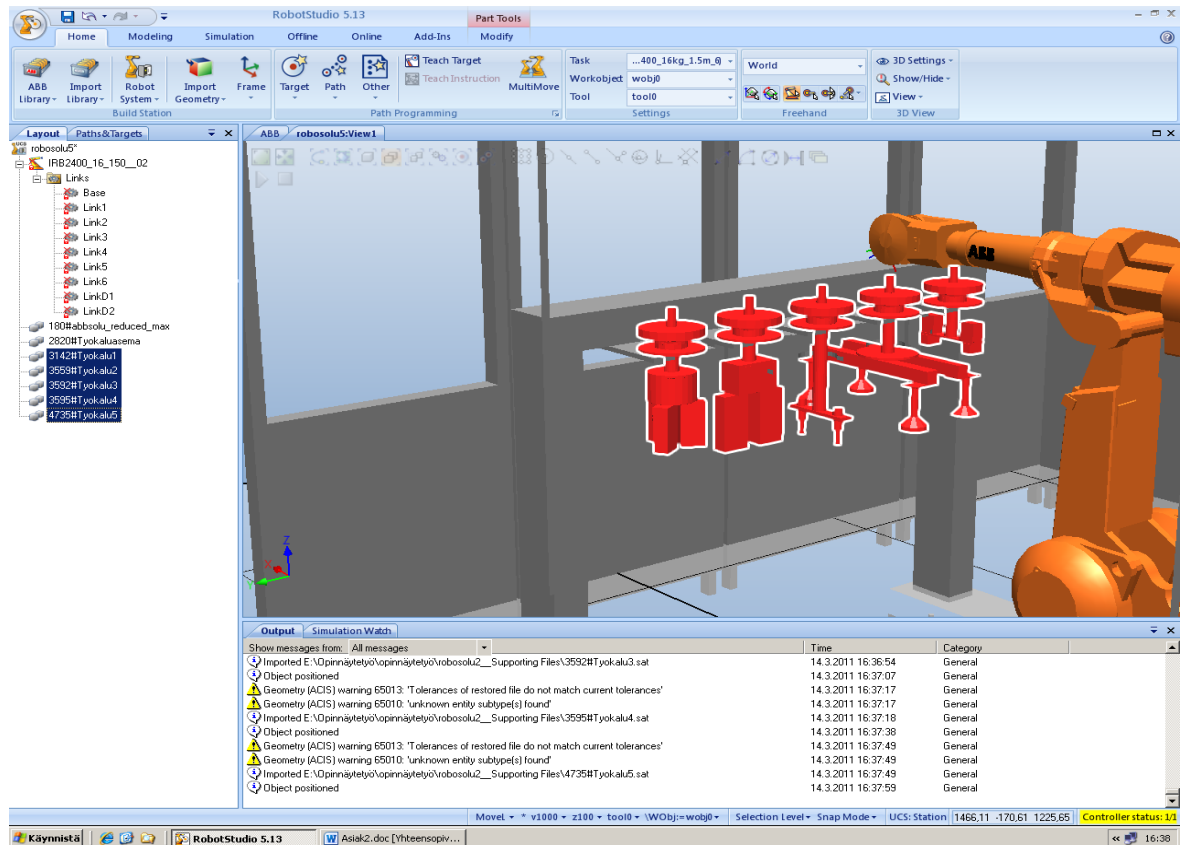
Kuva 28. Koordinaattien muuttaminen.

Seuraavaksi ohjelmaan tuotiin työkaluasema, joka on teline jossa sijaitsevat robotin työkalut. Set position -toiminnon avulla työkaluasema saatiin myös oikealle etäisyydelle robottiin nähden (Kuva 29).



Kuva 29. Työkaluaseman siirto.

Simuloidut robotin työkalut siirrettiin seuraavaksi ohjelmaan yksi kerrallaan. Jokainen työkalu piti myös asemoida työkaluasemaan oikealle paikalleen. Työkaluja on yhteensä viisi kappaletta (kuva 30).



Kuva 30. Robotin työkalut.

3.4 Työkalujen liittäminen robottiin

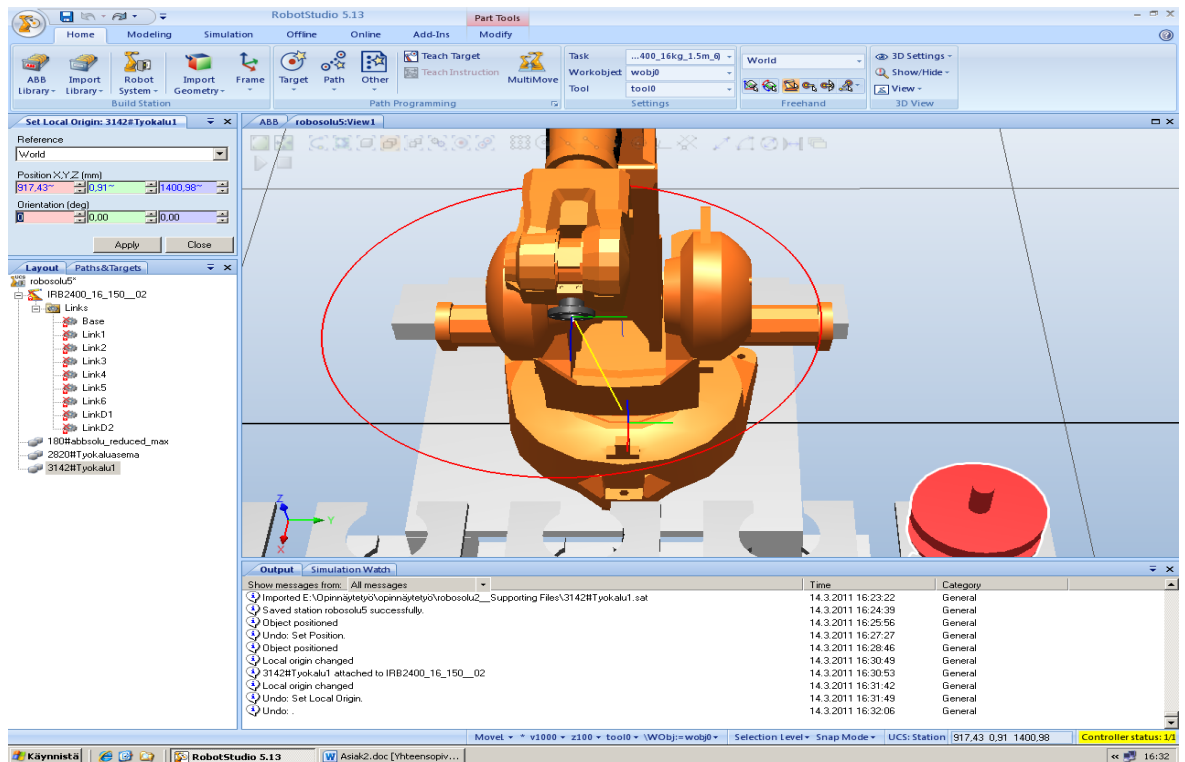
Opinnäytetyön tavoitteena oli myös saada liitettyä ohjelmaan tuodut työkalut robottiin. Työkalun on tarkoitus kiinnittyä robotin laippaan ja liikkua robotin liikkeiden mukana.

Työkalu voidaan kiinnittää robottiin kahdella tavalla. Voidaan valita haluttu työkalu hiiren vasemmalla näppäimellä ja raahata se pitämällä näppäintä pohjassa robotin nimen päälle layout-valikossa. Ohjelma kysyy tässä vaiheessa halutaanko valitun työkalun asema pitää nykyisenä, johon vastataan ei. Jos vastataan kyllä, työkalu jää työkaluasemaan, mutta liikkuu samalla etäisyydellä kun robottia liikutetaan. Toinen tapa on valita oikealla hiiren näppäimellä työkalu layout-

valikossa ja sen jälkeen avautuvasta valikosta vasemmalla hiiren näppäimellä valitaan toiminto Attach. Tämän jälkeen avautuu vielä uusi valikko, josta näkyvät solun osat eli robotti, seinät, työkaluasema ja työkalut. Valitaan robotti IRB2400_16kg_150_02. Ohjelma esittää tässä vaiheessa saman kysymyksen, kuin ensimmäisessä tavassa, eli pidetäänkö valitun työkalun asema nykyisenä. Vastataan tähän ei.

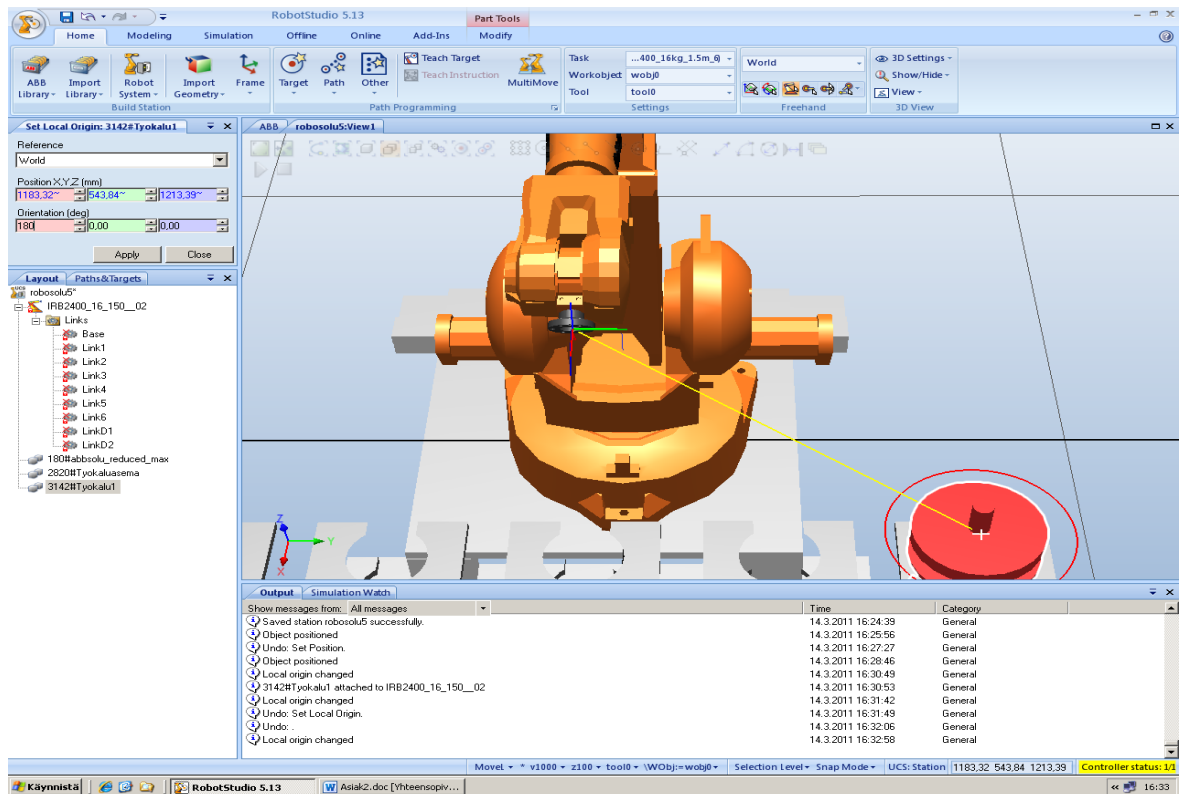
Nyt kun työkalun kiinnittäminen oli saatu valmiiksi, huomattiin, että työkalu ei ole kiinnittynyt oikein robotin laippaan vaan on suurella etäisyydellä robotista. Tämä johtuu siitä, että työkalun oma koordinaatisto ei ole oikeassa kohtaa robotin maailmankoordinaatiston keskipisteen suhteen. Työkalut piti asemoida robotin laipan koordinaattien mukaisesti. Tämä onnistui set local origin -toiminnon avulla.

Set local origin -toiminnolla asetettiin kappaleen oma koordinaatisto oikeaan asemaan. Tämä toiminto avattiin painamalla hiiren oikealla napilla haluttua kappaletta ja sen jälkeen avautuvasta valikosta valitsemalla set local origin. Tässä tapauksessa aluksi valittiin työkalu1. Ruudun vasempaan reunaan tuli näkyviin set local origin -valikko. Tämän jälkeen valittiin hiiren vasemmalla näppäimellä robotin laippa, johon työkalun oli tarkoitus kiinnittyä. Robotin ympärille tuli näkyviin punainen ympyrä ja keltainen viiva (kuva 31). Tässä vaiheessa Set local origin -valikosta valittiin komento Apply, jolla hyväksyttiin numeroarvot.



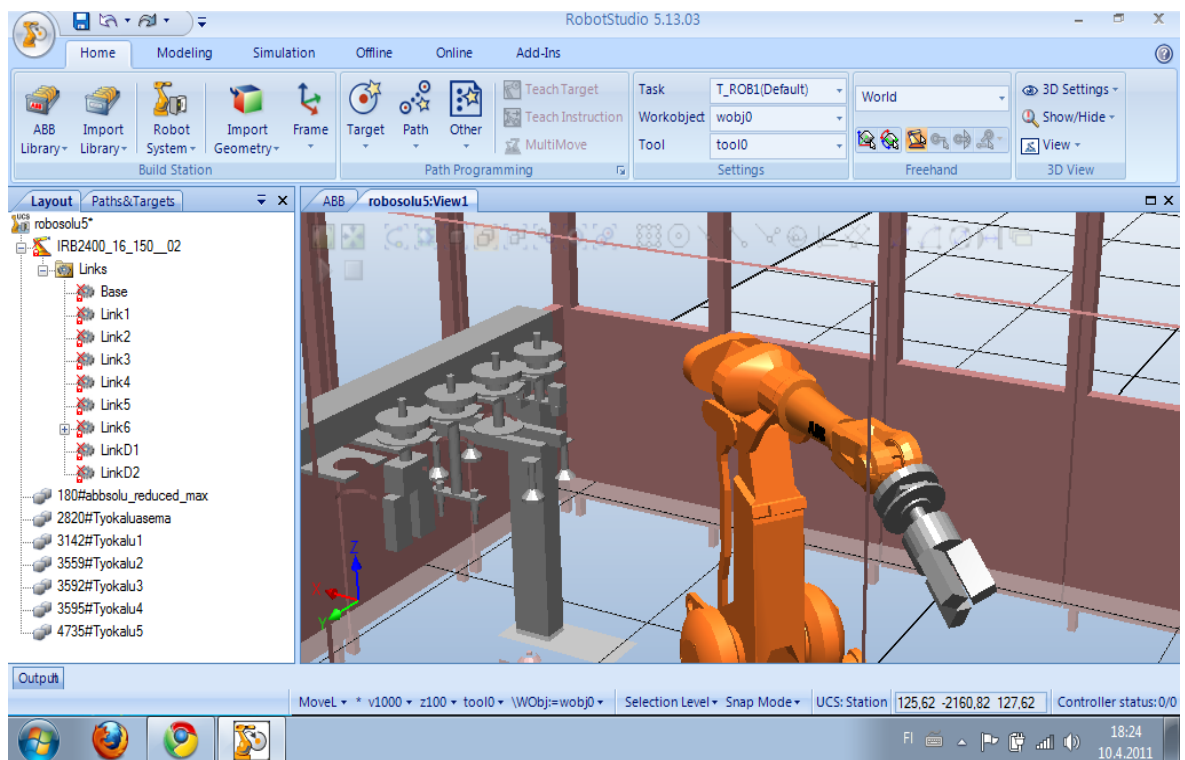
Kuva 31. Robotin laipan valinta set local origin -toiminnolla

Tämän jälkeen valittiin hiiren vasemmalla näppäimellä työkalun kiinnitysosa, jonka olisi tarkoitus kiinnittyä robotin laippaan. Robotin laipasta tuleva keltainen viiva siirtyi nyt osoittamaan työkalua (kuva 32). Sitten muutettiin set local origin -valikosta työkalun orientation-arvo eli astearvo lukemaan 180. Muuten työkalu kiinnittyy laippaan väärinpäin. Samasta set local origin -valikosta valittiin jälleen komento Apply.



Kuva 32. Robotin työkalun valinta set local origin –toiminnolla

Tämän jälkeen työkalu kiinnittyi robotin laippaan oikein (kuva 33).



Kuva 33. Työkalu kiinnitettynä robottiin

4 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli saada aikaan CAD-malli ABB:n robotin solusta, joka saataisiin vietyä RobotStudio-ohjelmaan ja hyödynnettyä opetuksessa. Työn tekemistä edisti Tapio Hellmanin mallintama CAD-kuva robotiikan laboratoriosta.

Työn suurimmat haasteet olivat CAD-geometrian siirtämisessä RobotStudioon. Koko alkuperäisen kuvan siirtäminen ohjelmaan huomattiin mahdottomaksi pintojen suuren määrän vuoksi. Tämän takia päädyttiin poistamaan kaikki työn kannalta turhat pinnat robottisolun ympäriltä. RobotStudio-ohjelman lisenssipuutteiden vuoksi solusta jouduttiin CAD-kääntäjällä tekemään useita versioita. Siirtäminen onnistui lopulta ACIS-tiedostomuodossa.

Työ edellytti syvällisempää perehtymistä RobotStudio-ohjelmaan. Ohjelman alkeet olivat tekijälle tuttuja robotiikan kurssilta.

Työn tavoite täyttyi siltä osin, että robottisolu saatiin siirrettyä ohjelmaan ja robotin työkalut saatiin liitettyä käsin työkalulaippaan kiinni. Kaikki solun osat on tallennettu kirjastotiedostoina ja voidaan liittää ohjelmaan myös yksittäin. Työn puutteena on sen toimimattomuus automaattiajossa eli robotti noutaisi työkalun itse työkaluasemasta.

LÄHTEET

Aalto, H. & Lylynoja, A. 2005. Robottien offline –ohjelmointiprojektin toteutus. [PDFdokumentti]. [Viitattu 25.4.2011]. Saatavissa:

http://www.delfoi.com/web/solutions/production/robotiikka/fi_FI/off-line/

ABB 2011. RobotStudio-ohjelmiston yleiskatsaus. [www-dokumentti]. ABB. [Viitattu 11.4.2011]. Saatavissa:

<http://www.abb.com/product/seitp327/a3aae74d6d9cb7dcc1257559003458bb.aspx?productLanguage=fi&country=FI>

ABB 2011. RobotStudio-työkalut. [www-dokumentti]. ABB. [Viitattu 11.4.2011] Saatavissa:

<http://www.abb.com/product/seitp327/a3aae74d6d9cb7dcc1257559003458bb.aspx?tabkey=4>

ABB 2011. IRB 2400. [www-dokumentti]. [Viitattu 11.4.2011]. ABB. Saatavissa:

<http://www.abb.fi/product/seitp327/e2c89161e03525ccc12572e6002e6e7a.aspx?productLanguage=fi&country=FI>

Aseko 2009. Servomootorit ja ohjaimet. [www-dokumentti]. Aseko Osakeyhtiö. [Viitattu 24.11.2010]. Saatavissa:

http://www.aseko.fi/index.php?group=00000736&mag_nr=3

CADblog 2010. Robot motion. [www-dokumentti]. CADblog.info. [Viitattu 24.11.2010]. Saatavissa: <http://cadblog.info/2011/03/03/robot-motion/>

Cimcorp 2010. MBR-robottitekniikka. [www-dokumentti]. Cimcorp Osakeyhtiö. [Viitattu 24.11.2010]. Saatavissa: <http://www.cimcorp.fi/MBR-robottitekniikka>

Edward Red 2011. Robotics Review. [www-dokumentti]. Robotics Review [Viitattu 25.4.2011]. Saatavissa:

http://research.et.byu.edu/eaal/html/RoboticsReview/body_robotics_review.html

Hellman T. 2006. Laboratorioinsinööri. Tekniikan yksikkö. Robottiikanlaboratorion simulaatio. 2000-2002. [www-dokumentti]. [Viitattu 21.11.2006]. Saatavissa: Rajoitettu saatavuus.

Hellman, T. <Tapio.Hellman@seamk.fi> 7.3.2006. Robotstudio [Henkilökohtainen sähköpostiviesti]. Vastaanottaja: Veli-Matti Pukkinen [Viitattu 21.11.2006]..

Keinänen, T. , Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOY Oppimateriaalit Oy.

Keinänen, T. , Kärkkäinen, P., Metso, T. & Putkonen, K. 2000. Logiikat ja ohjausjärjestelmät. Helsinki: WSOY Konetekniikka.

Kuivanen, R.1999. Robottiikka. Vantaa: Talentum

Kiowa 2010. SEC-AC-305-PB-P01 Festo motor controller. [www-dokumentti]. Kiowa Ltd. [Viitattu 24.11.2010]. Saatavissa: http://www.kiowa.co.uk/products/festo_motor_controller_sec-ac_for_servo_motors/EF533778

Malm, T. 2008. Vuorovaikutteisen robotiikan turvallisuus. Helsinki: Hakapaino Oy.

Penlink 2010. Pulssianturi. [www-dokumentti]. Penlink AB. [Viitattu 24.11.2010]. Saatavissa: <http://www.penlink.se/fi/produkter/pulsgivare>

RobotWorx 2011. ABB IRC5 Teach Pendant. [www-dokumentti]. RobotWorx. [Viitattu 24.11.2010]. Saatavissa: <http://www.robots.com/abb.php?controller=irc5>

Teollisuusrobottitilasto 2002. Helsinki: Suomen Robotiikkayhdistys ry. [PDFdokumentti]. [Viitattu 11.4.2011] Saatavissa: http://www.roboyhd.fi/index.php?option=com_docman&task=doc_download&gid=26&Itemid=66

Teollisuusrobottitilasto 2005. Helsinki: Suomen Robotiikkayhdistys ry. [PDFdokumentti]. [Viitattu 11.4.2011] Saatavissa: http://www.roboyhd.fi/index.php?option=com_docman&task=doc_download&gid=29&Itemid=66